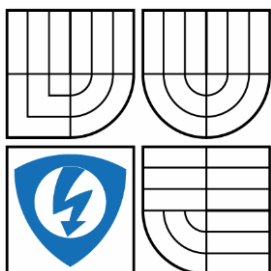


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH  
TECHNOLOGIÍ

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION  
DEPARTMENT OF CONTROL AND INSTRUMENTATION

## ŘÍDICÍ SYSTÉM MOBILNÍHO ROBOTU

MOBILE ROBOT CONTROL SYSTEM

### BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

### AUTOR PRÁCE

AUTHOR

Loránt Bugár

### VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Ondřej Hynčica

BRNO 2014



**VYSOKÉ UČENÍ  
TECHNICKÉ V BRNĚ**

**Fakulta elektrotechniky  
a komunikačních technologií**

**Ústav automatizace a měřicí techniky**

# Bakalářská práce

bakalářský studijní obor  
**Automatizační a měřicí technika**

**Student:** Loránt Bugár

**ID:** 145975

**Ročník:** 3

**Akademický rok:** 2013/2014

**NÁZEV TÉMATU:**

## Řídicí systém mobilního robotu

### POKYNY PRO VYPRACOVÁNÍ:

Navrhněte a realizujte řídicí systém dálkově ovládaného mobilního robotu. Řídicí systém bude využívat operační systém reálného času a bude implementován na vestavném mikrokontroléru s architekturou ARM.

- 1) Navrhněte, popište a realizujte jednotlivé části řídicího systému robotu. Využijte pokud možno hotové moduly a open-source řešení.
- 2) Proveďte identifikaci podvozku, navrhněte a implementujte regulátory motorů. Srovnajte identifikované a teoreticky získané výsledky.
- 3) K řídicímu systému připojte bezdrátový komunikační modul založený na Wi-Fi, který bude sloužit k přenosu povelů a dat.
- 4) Připojte snímače veličin prostředí (teplota, vlhkost, složení plynů apod.) a kameru. Měřené a obrazové údaje přenášejte do PC.
- 5) Vytvořte jednoduchou aplikaci pro PC, která bude sloužit k ovládání robotu a zobrazování přijatých dat.

### DOPORUČENÁ LITERATURA:

SLOSS, Andrew N, Dominic SYMES a Chris WRIGHT. ARM system developer's guide: designing and optimizing system software. Amsterdam: Elsevier, 2004, 689 s. ISBN 15-586-0874-5.

**Termín zadání:** 10.2.2014

**Termín odevzdání:** 26.5.2014

**Vedoucí práce:** Ing. Ondřej Hynčica

**Konzultanti bakalářské práce:**

**doc. Ing. Václav Jirsík, CSc.**

*Předseda oborové rady*

## **Abstrakt**

Bakalárska práca je zameraná na vytvorenie mobilného robota s operačným systémom reálneho času. Riadiací systém robota je založený na mikrokontroléri Arduino s procesorom ARM Cortex - M3. Praktická časť práce sa zaoberá vyriešením klient-server komunikácie s UDP protokolom, s identifikáciou motora z nameraných dát a následne návrhom PI regulátoru. Pozornosť v poslednej časti práce venujem riešeniu komunikácii so zbernicou I2C a používaniu senzorov: akcelerometer, ultrazvukový snímač, infra snímač, svetelný snímač, hall senzor, a používanie robotického ramena. Je vytvorená ukážková aplikácia na ovládanie robota a na vizualizácie senzorových hodnôt.

## **Klíčová slova**

ARM, Cortex M3, Arduino, L298, W5100, operačný systém reálneho času, RTOS, PWM, PID, regulátor, IIC, I2C, akcelerometer, hall senzor, DC motor, akcelerometer, ultrazvukový snímač, infra snímač, svetelný snímač, hall senzor, robotické rameno.

## **Abstract**

Bachelors project is fixated for creating a mobile robot with a real time operating system. The robot control system is based on a microcontroller Arduino with ARM Cortex-m3 processor. The practical part is dealing with resolving the client-server communication with the UDP protocol, also with the identification of the engine from measured data's by the Simulink and subsequently designing the PI regulator from the identified data's. In the last part of the work the focus is giving to solving the communication with I2C bus and using sensors like : accelerometer, ultrasonic sensor, infra sensor, light sensor, hall sensor and usage of a robotic arm. An application is created for controlling the robot and for visualisation of sensor readings..

## **Keywords**

ARM Cortex M3, Arduino, L298, W5100, Real Time, RTOS, PWM, PID controller, IIC, I2C, accelerometer, Hall sensor, DC motor, accelerometer, ultrasonic sensor, infra sensor, light meter, hall sensor, robotic arm.

**Bibliografická citace:**

BUGÁR, L. Řídicí systém mobilního robotu. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2013. 43s. Vedoucí bakalářské práce byl Ing. Ondřej Hynčica.

## **Prohlášení**

„Prohlašuji, že svou bakalářskou práci na téma Řídicí systém mobilního robota jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne: **26. května 2014**

.....  
podpis autora

## **Poděkování**

Děkuji vedoucímu bakalářské práce Ing. Ondřeji Hynčicu za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé bakalářské práce.

V Brně dne: **26. května 2014**

.....

podpis autora

# Obsah

1	Úvod .....	8
1.1	Myšlienka .....	8
2	Teoritický rozbor .....	9
2.1	Telepresence .....	9
2.2	Arduino .....	9
2.3	ARM Cortex M-3 .....	12
2.4	Operačný systém reálneho času .....	13
3	Riešenie .....	14
3.1	Architektúra robota .....	14
3.2	Komunikácia .....	15
3.2.1	UDP protokol .....	15
3.3	Regulácia .....	16
3.3.1	Meranie otáčky .....	16
3.3.2	Identifikácia .....	17
3.3.3	Návrh regulátoru .....	19
3.3.4	Ovládanie motora .....	22
3.4	Operačný systém reálneho času .....	23
3.4.1	Tasky .....	24
3.4.2	Meranie vlákna .....	26
3.5	Senzory .....	27
3.6	Digitálne senzory .....	27
3.6.1	Komunikácia I2C .....	27
3.7	Analogové sensory .....	31
3.8	Kamera .....	31
3.9	Robotické rameno .....	32
3.10	Napájanie .....	33
3.11	PC aplikácia .....	34
3.12	Shrnutie .....	35
4	Záver .....	36

# 1 ÚVOD

Robotika je v dnešnej dobe veľmi populárna, používajú skoro u každej vetve výroby, medicíny alebo v armáde. Každá oblasť má svoju špecifickú problematiku, ktorú sa snažia za pomoci robotov vyriešiť. Spektrum problematiky je veľmi široké, preto je potrebné stále vyvíjať nové druhy a typy robotov.

V tomto projekte bude navrhovaný a vyrobený teleprezenčný robot, ktorý bude merať fyzikálne veličiny v jeho okolí. Bude riešený obojsmerný klient – server komunikácia s UDP protokolom. Motory pomocou hallového senzoru budú identifikované v Matlabe pomocou - identification toolbox. Pre motory z identifikované hodnoty bude navrhovaný regulátor, ktorý bude nastavený pomocou -Matlab PID toolbox.

V ďalšej časti práce bude riešená komunikácia a spracovanie hodnoty od senzorov. Bude riešené riadenie servo pohonov pre robotické rameno a pre pohyb s kamerou. Dáta od senzorov budú použité na pomáhanie riadenia robota. Riadiaci algoritmus bude rozdelený na „tasky“ pomocou RTOS a budú sa cyklicky opakovať v požadovanom intervale. Z strane počítača bude realizovaný program pre riadenie robota, riadenie robotického ramena, monitorovanie meraných hodnôt a sledovanie obrazu z kamery.

## 1.1 Myšlienka

V armáde sa často používajú teleprezenčné roboty pri katastrofách alebo pre preskúmanie daného okolia. Nami navrhovaný teleprezečný robot bude slúžiť pre meranie okolia za pomoci rôznych snímačov uložených na konštrukcii robota. Celý robot sa skladá: z podvozku, z manipulátoru, z routera, z riadiaceho embedded systému a zo snímačov.

Router slúži na bezdrôtovú komunikáciu s robotom cez Wi-fi. Na router je napojený s káblom embedded systém a IP-kamera. Emedded systém priamo dostane príkazy od počítača cez router, ktoré následne vyhodnotí, a podľa toho reaguje. Na embedded systému sú napojené senzory za pomoci ktorých meria rôzne veličiny. Aby embedded systém na všetko reagoval, a všetko plynule meral, je nainštalovaný doňho operačný systém reálneho času, s ktorým sú jednotlivé úlohy rozdelené tak, že vytvorí multitasking. Z toho vyplýva, že vykoná teoretický naraz viac úloh. Praktický embedded systém nemá viac jadrá a preto nemá paralelný spracovanie, ale pomocou RTOS sa dá vytvoriť multitasking, tak že úlohy môžeme rozdeliť na vlákna a potom jednotlivé úlohy prepína.



## 2 TEORITICKÝ ROZBOR

### 2.1 Telepresence

Telepresence je spôsob pozorovania reálneho života tak že pozorovateľ je fyzicky inde a z diaľkovo ovláda taký stroj, ktorý má ľudské schopnosti ako videnie alebo počúvanie. Často tieto stroje majú senzory, manipulátory. Schopnosť manipulovať objektov je vykonaný vo veľkom množstve spôsobov, podľa závislosti na potrebách užívateľa. Typický, používaný manipulátor je model ľudského ruka, ktorý riadia servá. Robot vo vzdialenom mieste skopíruje ľudské pohyby, ktoré operátor vykoná. Táto schopnosť sa nazýva teleoperation.[3]

Teleprezenčný roboty môžu byť použité v nebezpečných situáciách napríklad pri pyrotechniku, pri vojenskej operácie, záchranu obetí z ohňa, toxické prostredie alebo v rukojemných situácie. Teleprezenčné roboty tiež používajú pri skúmanie iných svetov, ako je v Mars Rovers, ktorý je teleoperačný zo Zeme. [3]



1. Obrázok: Teleprezenčný robot v U.S armade [2 ]



2. Obrázok : Teleprezenčný robot v čínskom armáde [1 ]

### 2.2 Arduino

V tejto projekte riadiaci systém je založený na Arduino.

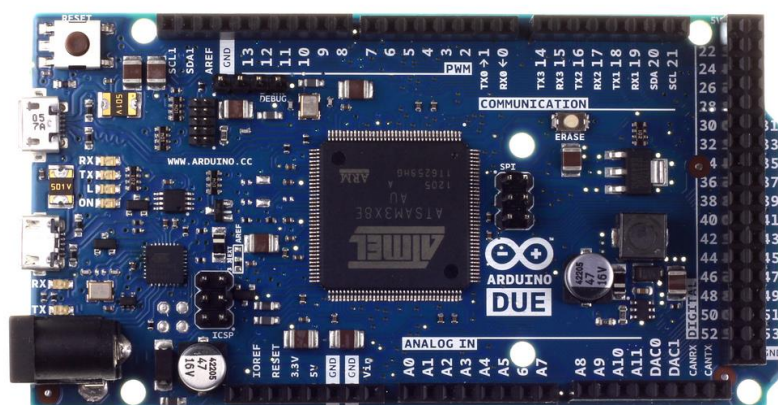
Arduino je open – source platforma je založená na mikrokontroléru Atmega od firmy Atmel. Projekt vznikol v roku 2005 v Taliansku. Jeho cieľom bolo vytvoriť prototypovací platformu pre študentov, ktorá umožní vývoj nových projektov. Projekt zaznamenal veľký úspech a neskôr začali vznikať jeho ďalšie, novšie verzie.

Arduino je najpopulárnejší embedded platforma na svete vďaka k tomu, že vyrábajú veľa príslušenstva čo veľmi uľahčuje prácu pri vývoji.

Prvá arduinová čísla boli na trhu sa vyrábali len Atmel mikrokontrolérom a nemali také dobré parametre, ako niektoré iné embedded systémy. Mali 8 bitu a 16 MHz, ale aj

tak zaznamenali veľký úspech lebo projekt Arduino je už od svojho počiatku voľne dostupné pre všetkým užívateľom, ktorí ho chcú používať alebo vylepšovať. Dokumentácia a referenčná príručka jazyka a externých knižníc je potom väčšinou vydávaná pod licenciou Creative Commons . Vďaka tomu sú k dispozícii všetky zdrojové súbory a je rešpektovaná určitá kompatibilita ( či už pomocou jazyka Java , v ktorom je napísané vývojové prostredie ( IDE ), alebo vďaka použitiu USB - to - RS232 prevodníkového chipu, ktorý zaisťuje virtuálne sériovú linku, a pre ktorý sú bežne dostupné ovládače na väčšinu platforiem ) .

Arduino s 8 bitovým processory boli vymyslené pre malé projekty a na pár senzorov. V dnešnej dobe, keď už aj telefóny majú 32 bitové procesory a má ju skoro každá konkurencia, ako napríklad Texas Instruments alebo Freescale embedded platformy, musel aj Arduino navrhnuť novšie silnejšie dosky. [5]

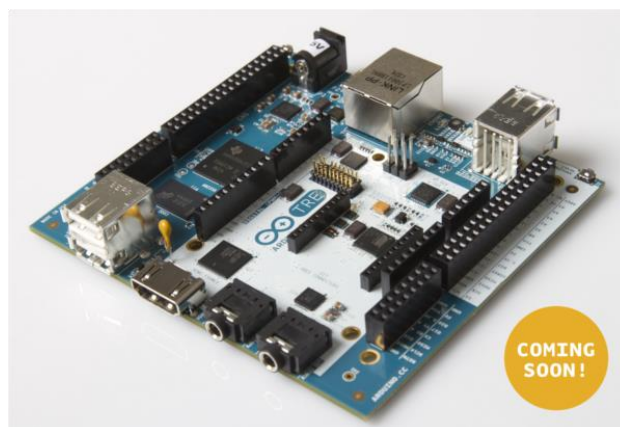


3. Obrázok: Arduino DUE s ARM M3 procesorom [4]

V 2013 Arduino vydal DUE ktorý má nový processor od ARM, Cortex – M3 a rýchlosť zvýšil na 84 MHz, 100MIPS. V lete 2014 sa chystá vydať ešte 2 nové platformy Galileo a TRE . Galileo dostane Intel processor X1000 a TRE, ktorý bude hybridný - bude mať processor ARM Cortex-A8 a Atmega32.



5. Obrázok: Arduino Galileo s Intel procesorom [4]



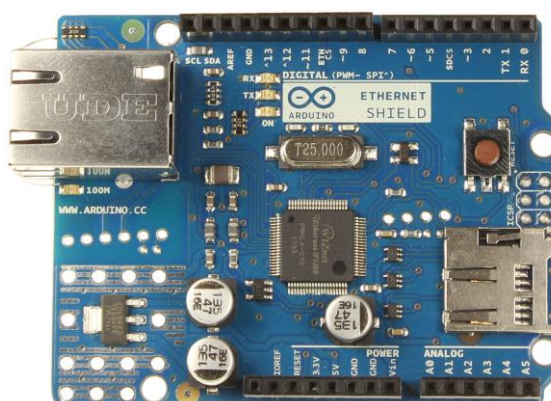
4. Obrázok : Arduino TRE [4]

Každá verzia je kompatibilná s príslušenstvami (shield), ktorý vyrába Arduino. Aj tie shieldy sú open source, takže na nete je ich nespočetné množstvo klonovaných, prerobené shieldy pre špecifické účeli.

V tomto projekte je používaný Arduino DUE a Ethernet Shield. Shield s chipom W5100 pre komunikáciu cez RJ-45, ktorý dokáže 100Mb/s a má aj rozhranie čítanie/písanie na pamäťovú kartu, napríklad pre učeli zálohovanie. Preto bol vybraný Arduino DUE lebo bol potrebný 32 bitový procesor, kvôli viac senzoru a kvôli RTOS. Menšie Arduiná majú 8 bitový procesor a pri viac senzoru a s RTOS by nezvládli spracovávať. Sú potrebné aj čím viac digitálnych vývodou, a je veľká výhoda, keď podporuje shieldy ako ethernet shield, alebo motor shield a to všetko zabezpečuje Arduino DUE. Väčšie embedded systémy ako Raspberry PI, alebo iné komplexné systémy boli by zbytočné.

	Microcontroller	Clock Speed	Digital I/O Pins	Analog Input Pins	Flash Memory	Cena
Arduino Nano	Atmega 168 /328	16 MHz	14	8	16 / 32 kB	33 \$
Arduino Micro	ATmega32u4	16 MHz	20	12	32 kB	16 \$
Arduino Mini 05	ATmega328	16 MHz	14	8	32 kB	12 \$
Arduino UNO Rev3	ATmega328	16 MHz	14	6	32 kB	20 \$
Arduino Leonardo	ATmega32u4	16 MHz	20	12	32 kB	18 \$
Arduino YÚN WI-FI	ATmega 32u4	16 MHz	20	12	32 kB	52 \$
Arduino Mega2560 Rev3	ATmega2560	16 MHz	54	16	256 kB	39 \$
Arduino DUE	Atmel SAM3X8E	84 MHz	54	12	512 kB	36 \$

1. Tabuľka: Typy arduiná [4]



6. Obrázok: Arduino ethernet shield [4]



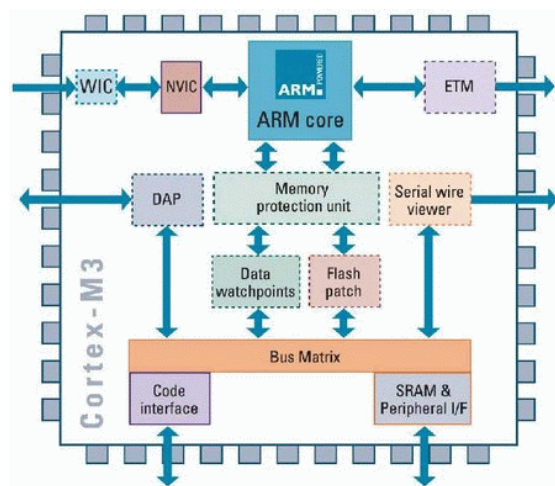
## 2.3 ARM Cortex M-3

Rodina procesorov založených na architektúre ARM Cortex poskytuje potrebné zdroje na riešenie aj náročných priemyselných, automobilových, bezdrôtových a mnohých ďalších technológií. Rodina obvodov je zložená z troch hlavných typov architektúr: ARMv7 je typ pre veľmi náročné aplikácie, bežiacie na zložitých operačných systémoch. Typ R je určený pre použitie v real - time systémoch a typ M je naproti tomu optimalizovaný pre použitie v lacných embedded aplikáciách. [6]

Rodiny procesorov Cortex - M3 sú prvé procesory architektúry ARMv7 - M , ktoré boli navrhnuté s cieľom dosiahnuť vysokého výkonu systému a maximálnej efektivity lacných vstavaných aplikácií , ako sú napríklad priemyselné riadiace systémy , automobilová elektronika , drôtové a bezdrôtové telekomunikačné systémy , systémy elektrického ovládania atď. na dosiahnutie tohto cieľa však bolo potrebné vykonať v základnej architektúre rad zásadných zmien . Najmä veľmi zjednodušiť proces tvorby softvérového balíka . Výsledkom je cenovo efektívne využitie procesorov architektúry Cortex - M3 a to aj v tých najjednoduchších aplikácií .

Procesory Cortex -M3 v závislosti na použitej Harvardskej technológii s tromi fázami spracovania dát ( pipeline ), ktoré ponúka rad príležitostí. Množstvo jedno - cyklových operácií prináša až 1,25 MIPS / MHz ( v praktickom teste ). V procesoroch Cortex - M3 je k dispozícii nová sada príkazov s označením Thumb - 2, ktorá programátorovi umožňuje dosiahnutie až o 70 % vyššieho výkonu na jeden megahertzov než klasické ARM procesory a umožňuje rýchlejšiu odozvu na prerušenie. [6]

Jádro mikrokontroléru umožňuje ladenie viac spôsobom ako JTAG a SWD JTAG cez blok DAP (Debug Access Port) pre sledovanie premenné používa Data watchpoints a breakpoints bloky. Jádro obsahuje aj MPU (Memory protection unit) ktorý je pre ochranu pamäti. Jádro môže pracovať vo dvoch režimoch. Režimu Thread, v ktorom processor vykoná kód aplikácii a v režimu Handler, vtedy vykonáva prerušenie. Obsluhu prerušenie zabezpečí blok NVIC (Nested Vector Interrupt Controller). Jádro obsahuje ešte 24-bitový časovač. [7], [8]



7. Obrázok: : Bloková schéma jadra ARM Cortex - M3 [6]

## 2.4 Operačný systém reálneho času

Režim reálneho času je taký režim, v ktorom aplikačné programy počítačového riadiaceho členu musia byť schopné stále spracovávať nové udalosti, prichádzajúce z riadeného procesu, pričom spracovanie udalosti a odpovedajúce reakcia musí nastať do určitého, vopred definovaného časového limitu.[9]

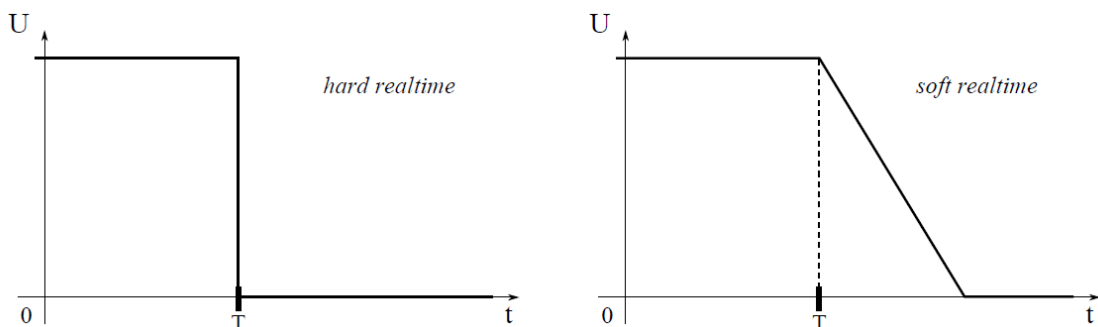
Operačné systémy reálneho času sa bežne vyskytujú a používajú v robotike, fotoaparátoch, komunikácie a má rôzne vojenské a vládne použitie. Embedded operačné systémy sú uložené zvyčajne v ROM chipe namiesto pevného disku a obsahuje iba absolútne potrebné súbory pre spustenie. [8]

V embedded systémoch sú väčšinou jednojadrové procesory, a používajú najčastejšie dve typu multitaskingu: kooperatívne a preemptívne - z hľadiska toho, kedy je možné uskutočniť prepnutie úlohy. U kooperatívneho multitaskingu, prepnutie úlohu riadi bežiacia úloha a u preemptívneho multitaskingu, prepnutie jednotlivé úlohy zaisťuje kód jadra operačného systému, ktorý je spustený periodicky a nezávisle na aktuálne bežiaci úlohe. Rozhodovanie, ktorá úloha bude nasledovať, zaisťuje plánovač jadra podľa prioritného plánovania, alebo podľa pravidelného prepnutia úlohy. U prioritného plánovania sa vykoná skôr tá úloha, ktorá má vyššiu prioritu. [8]

U pravidelnému prepínanie úloh je nadefinovaný vopred stanovený čas, a pri vypršania času úlohu prepína. U preemptívnych algoritmov môže dôjsť k prepnutiu v akejkol'vek časti spracovávanej úlohy. Rozdelenie riadiaci aplikáciu na viac úloh je veľmi uľahčuje prácu pri zložitejších systémoch. Rozdelené úlohy bežia nezávisle na druhé a tak sa dá ľahšie ladiť časti aplikácie. [8]

Operačné systémy reálneho času sú také operačné systémy, ktoré sú schopné reagovať na externé a interné udalosti v časovo deterministických úsekoch a pritom vykonávať ďalšie riadiace funkcie. Operačné systémy reálneho času sa ďalej delíme na soft real – time a na hard real – time systémy, podľa požiadavku na deadlines. [8]

Pri systémoch "real-time" musí operačný systém poznať maximálny čas pre každú kritickú operáciu. Operačné systémy, u ktorého požiadavky na stanovenie času reakcie sú absolútne nazývame "hard real-time" systémy. Systém je "hard-real time", ak by prípadné nedodržanie časových limitov malo katastrofálne následky. Operačné systémy, ktoré sú označené ako "soft real time" dovoľujú drobné odchýlky v reakciách. [27]

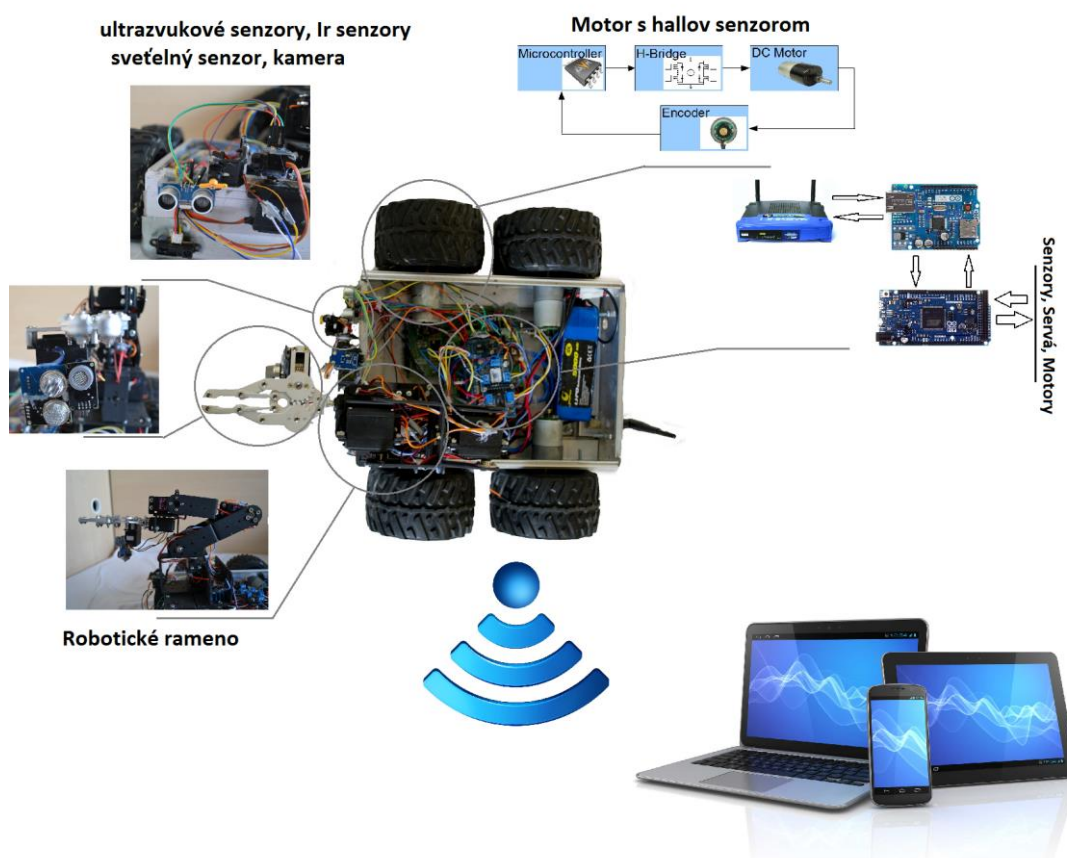


8. Obrázok: Užitočnosti hard a soft realtime systému [8]

## 3 RIEŠENIE

### 3.1 Architektúra robota

Robot skláda: Z routera, z Arduina, z snímače, z manipulátoru a z podvozku. Ako už viššie bol písané router slúži na komunikáciu s robotom cez Wi-Fi. Je to veľká výhoda lebo vďaka tomu že Wi-Fi je štandardizovaná ľahko sa dá nato aj pripojiť s iným elektronickým zariadením ako telefón alebo tablet lebo napríklad pri Xbee to by bol ťažko realizovateľný a tak by mal obmedzenie. A druhá veľká výhoda že má oveľa silnejší signál než ostatné vysielacie/prijímacie ako napríklad Bluetooth alebo Wi-Fi modul. Na router ľahko sa dá pripojiť IP kameru a embedded systém netreba zaťažovať spracovávaním obrazu lebo stará o tom IP kamera. Má i aj nevýhody a to že má pomerne veľkú spotrebu a veľkosť. Na router je napojený s káblom aj Arduino s pomocou ethernet shieldu čo priamo dostane príkazy od počítača cez router. Na embedded systéme sú napojené senzory z ktorými merá rôzne veličiny. Na poslanie príkazov slúži java aplikácia ktorá zabezpečí spolu s Arduino obojsmerná komunikácia a vizualizácia dáta.



9. Obrázok: Architektúra robota [33],[34]

## 3.2 Komunikácia

Základom komunikácie je metóda klient - server, ktorý rozlišuje systémy klienta od systémov servera, ktoré komunikujú cez počítačovú sieť. Pri klient - server architektúre je používaný systém, ktorý sa skladá zo softvéru klienta a zo softvéru servera. Softvér alebo proces klienta môže inicializovať komunikačné spojenie, zatiaľ čo server čaká na požiadavku od klienta. V tomto prípade server bude Arduino DUE a klient bude program v počítači, s ktorým ovláda robot.

Klient/server opisuje vzťah medzi dvoma počítačovými programami, z ktorých jeden je klient, odošle požiadavku na službu z druhého programu servera, ktorá požiadavku splní. Tento model sa stal centrálnou ideou sieťovania. [9]

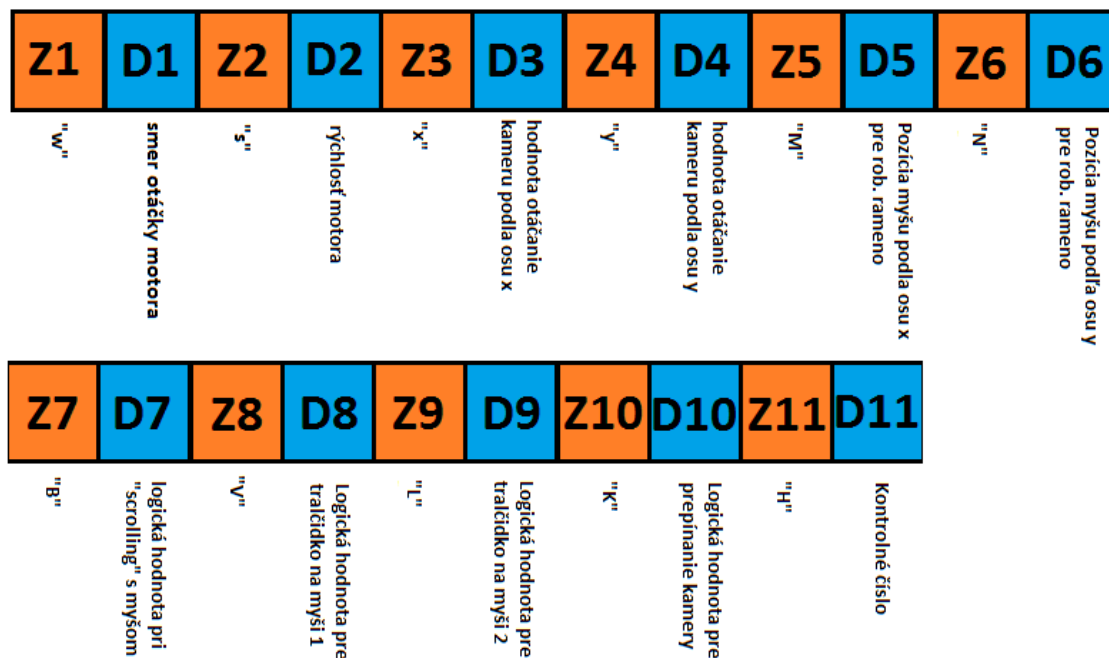
Robot sa skladá z Wi-Fi routera, ktorý slúži nato, aby sa dalo komunikovať bezdrôtovo, s protokolom 802.11b/g a max rýchlosťou 54Mbps. Router je napojený na Arduino s Ethernet shieldom a pripojení s UTP káblom. Technológia routera je pomerne stará, priemerná rýchlosť Wi-Fi 802.11g je 19-20 Mbit/s ale našom prípade je potrebná šírka pásma postačujúca. Bolo by oveľa lepšie používať WI-FI shield ako napríklad WiShield 3.0 [47], alebo nejaké iné rozhranie na komunikáciu, nakoľko router je veľký a má veľkú spotrebu. V robote je umiestnená aj IP kamera, čo je tiež napojená na router, pričom takto napojená kamera je dostupná. V praxi to znamená, že nie je potrebné obraz spracovávať a ani použiť iné komunikačné rozhranie. V našom prípade je pre naše účely lepším riešením použitie routera. [10]

### 3.2.1 UDP protokol

Komunikačný protokol bol zvolený ako UDP. Protokol posiela datagramy medzi počítačmi v sieti, ale na rozdiel od iných protokolov nezaručuje, že prenášané pakety sa nestratia a ani to že niektoré pakety neprídu viackrát. Napriek tomu používajú tento protokol, lebo je veľmi rýchly a jednoduchý tam, kde nebude vadiť, že nevieme skontrolovať či paket dorazil do cieľa alebo nie. UDP je veľmi ziskový, lebo výmenou dostaneme veľkú rýchlosť. Používajú napríklad streamované media. V tomto projekte je dôležité, aby bolo čo najmenšie meškanie a preto bol zvolený UDP protokol. Zabezpečí elimináciu meškania a kvôli stálej komunikácii nebude vadiť, keď niektoré pakety nebudú doručené a kvôli spôsobu spracovania informácie je zabezpečený aby na komunikáciu nebol vplyv pomiešané, polovine doručené pakety.

Komunikácia je s formou posielanie a prijímanie stringy. Klientsky software posiela string vo forme "Začínajúci znak, smer, koncový znak, ,rýchlosť, koncový znak,..." napríklad ("W,1,S, 300, x, 90, C, 90, M..."). Výhoda stringu je, že ľahko rozširovateľná a ľahko sa dá s ňým pracovať. V Arduine sa uloží string do matice, takzvané do packetbufferu a postupne sa spracováva. Spracované príkazy sa postupne uložia do premennej a jednotlivé vlákna vykonávajú zadané príkazy podľa premennej.

Počas vykonávania požadovaných zmien paralelne meria so senzormi, pričom namerané hodnoty ukladá do premenných. V predom nadefinovanom čase pravidelne posiela naspäť namerané hodnoty pre klienta formou stringu.



10. Obrázok: Komunikačný diagram. Zx-znak, Dx-Data

### 3.3 Regulácia

#### 3.3.1 Meranie otáčky

Na meranie rýchlosti motora sú použité zabudované hall senzory, umiestnené na motoroch. Každý motor má aj vlastnú prevodovku, preto je potrebné namerané hodnoty prepočítať. Je to nutné pre získanie skutočných otáčok kolesa.

Elektromotory sú jednosmerný s permanentným magnetom a s kovovou prevodovkou pomeru 29:1 a s integrovanou kvadrturným enkodérom, ktorý poskytuje rozlíšenie 64 impulzov na otáčku hriadeľa motora. Zodpovedá to 1856 krokov na otáčku výstupného hriadeľa prevodovky. Keby boli počítané klesajúce aj rastúce hrany A a B výstupu hallov senzoru, mohli by sme získať 128 impulzov na otáčku hriadeľa motora. Z frekvencie motora môžeme zistiť otáčky, a z presunutí dvoch signálov môžeme zistiť smer. V tomto prípade stačí použiť len jeden kanál, nakoľko smer otáčania motora vieme vopred. Keď sledujeme len jednu hranu na jednom kanále tak dostaneme 16 krokov na otáčku hriadeľa motora. Motory majú 350 otáčok za minútu pri 12V a 0.78 Nm momentu.



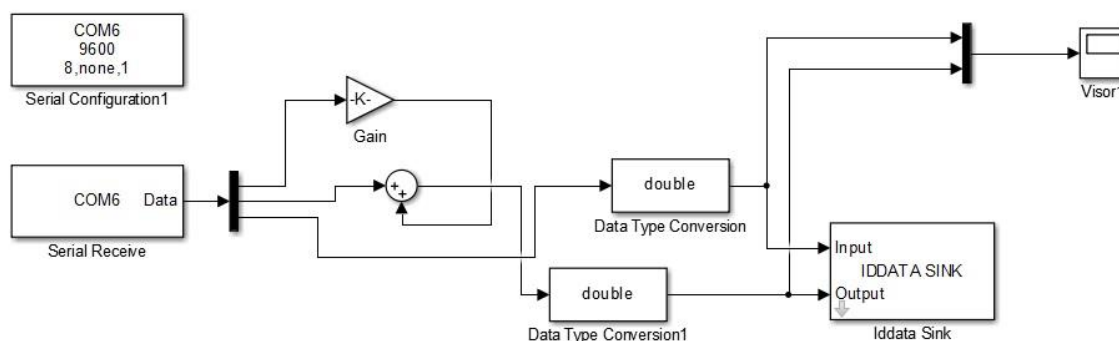
Výpočet skutočnej rýchlosti v algoritme je trochu zložitejší. Jeden kanál je používaný na prerušenie. Keď príde impulz program preskočí do jednej funkcie, kde premenná count inkrementuje alebo dekrementuje podľa toho, že impulz je akom logickom úrovni. Každých 100ms je volaná druhá funkcia, kde odčíta z count prechodný nameraný počet impulzov. Count nikdy nie je vynulovaný. Výsledok ešte treba prepočítať na minúty a celú hodnotu treba ešte rozdeliť s 464. Je to kvôli prevodovke, lebo prevodovka má 29:1 pomer - to znamená, že keď sa motor otočí 29 – krát, vtedy sa nám otočí koleso jedenkrát na prevodovke. Z jednej otáčky z motora dostaneme 16 impulzov, takže  $16 \cdot 29$ . Po prepočítaní dostaneme výsledok v RPM. [11],[43]

$$\text{Rýchlosť} = \left\lfloor \frac{(\text{Count} - \text{Countpredchozi}) \cdot (60 \cdot (1000 / \text{looptime}))}{16 \cdot 29} \right\rfloor \quad [\text{RPM}] \quad (3.1)$$

### 3.3.2 Identifikácia

Identifikácia motoru bol vyriešený pomocou Identification toolboxom. Hodnoty boli namerané bez i s podvozkom. Pre Simulink sa objavila knihovňa [36], [37] pre podporu arduina, ktorý pridá do simulinku nové moduly a menu na nahrávanie do chipu. Pomocou nich sa dajú nadráťovať úlohy, priamo programovať arduino, ovládať alebo len čítať poslané/prijímané dáta.

Pomocou knihovny bola vyriešená klasická 8-bitová sériová komunikácia medzi Simulinkom a Arduino. Do Arduino bol napísaný program pre cyklicky riadenie motora s PWM signálom, takže h - mostík dostal maximálnu hodnotu PWM signálu a po 5 s minimálnu hodnotu PWM signálu. Takto vieme zistiť, ako motor a h - mostík spolu reagujú na PWM. Jednotka času, ktorá bola nastavená na zmenu hodnoty PWM signálu bol navrhovaný tak, aby keď PWM signál dosiahne maximálnu hodnotu, motor mal dostatok času dosiahnuť konštantné otáčky a keď PWM je 0, aby motor mal dost času úplne zastaviť.

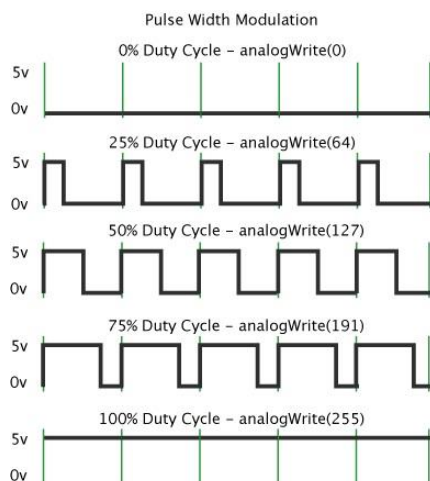


11. Obrázok: Diagram komunikáciu medzi Simulinkom a Arduino

Program v arduine počas riadenia motora počíta otáčky a každý 100 ms pošle hodnotu výstupu motora a hodnotu vstupu cez serial port. V Simulinku bol nadráťovaný

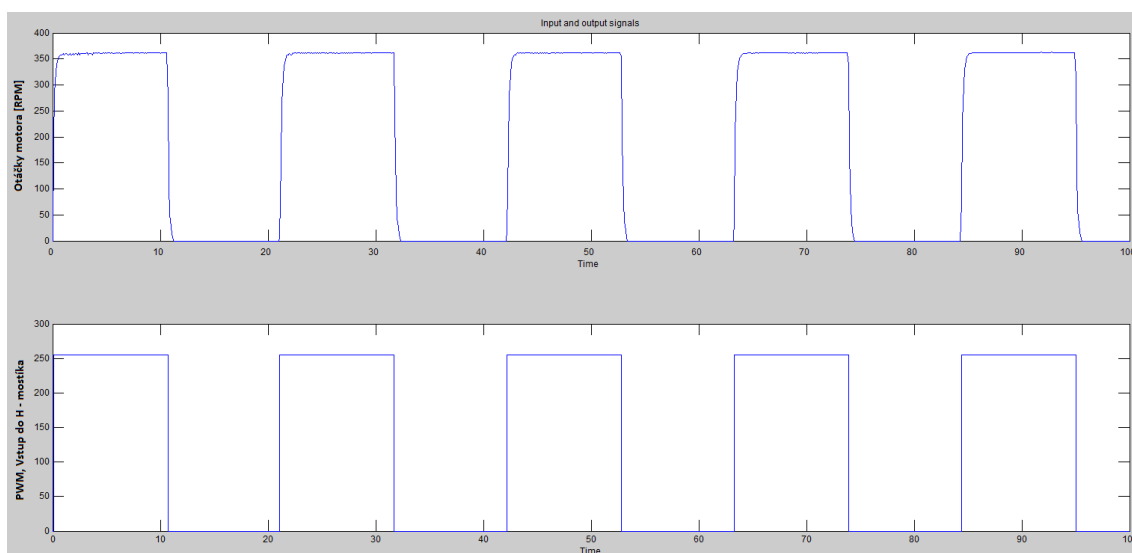
a nastavený seriálna komunikácia a uloženie vyčítané hodnoty vo formáte Time domain data (IDDATA) kvôli ďalšiemu spracovávaní.

U arduino PWM signál rozdelený na 255 hodnôt. Je to úmerne na skutočný signál ako je to aj názorné na obr. č.12. Takto je oveľa lepšie vidieť zmeny signálu pri znázornení na grafoch i aj pri ladení vhodnejší.



12. Obrázok: Skutočný PWM signál arduino a pomer podľa čísel [12]

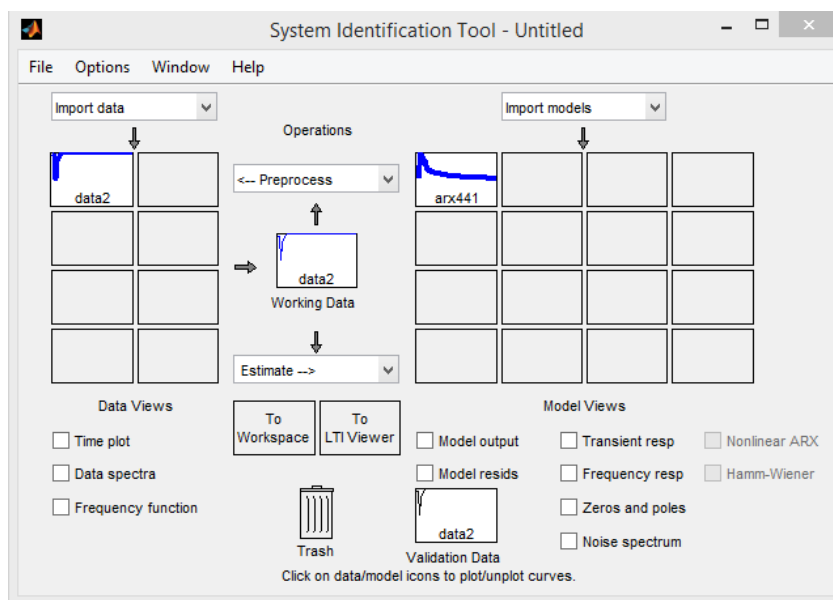
Seriálna komunikácia má obmedzenie a preto maximálna hodnota, čo ešte pošle bez overflow je 255, čo ešte pre PWM postačuje. U výstupných hodnôt motora by nám to nestačilo, nakoľko hodnoty sú od 0 do 350. Pre vyriešenie tejto problematiky treba rozdeliť väčšie číslo než 255 a poslať to v 2 bajtoch. Ďalej na strane host'a dať číslo znova dokopy



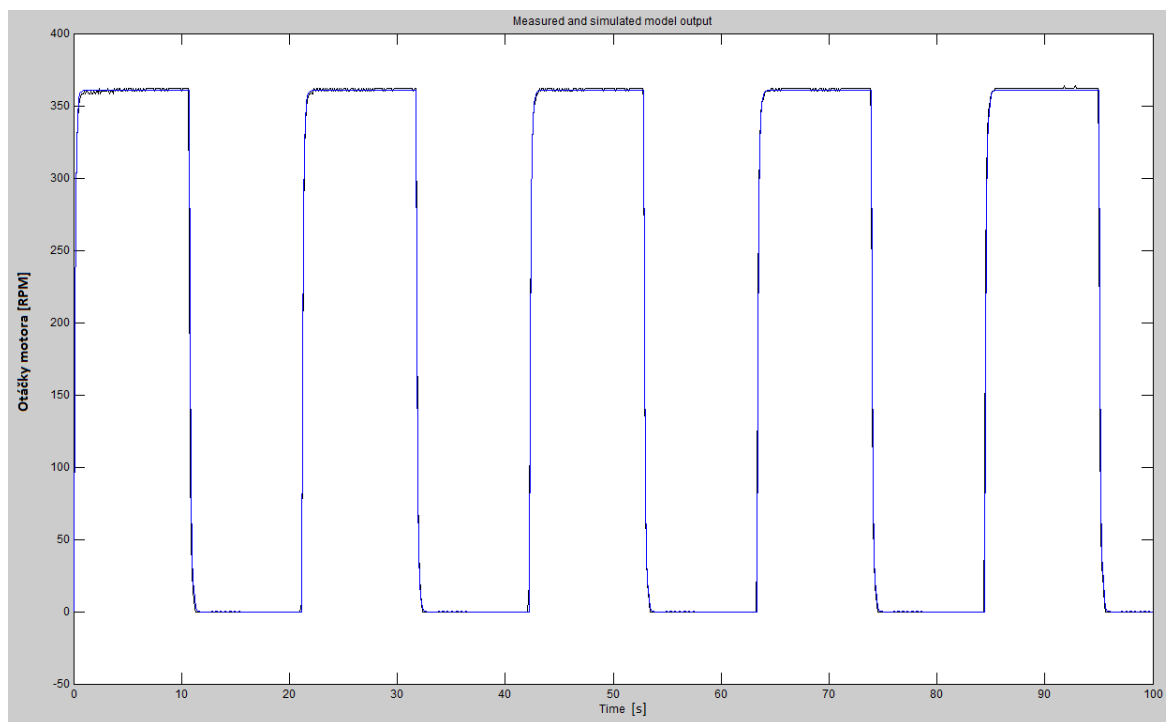
13. Obrázok: Reakcia motora na PWM signál. V hornom grafe sú otáčky motora a dolnom je pomer PWM signálu

### 3.3.3 Návrh regulátoru

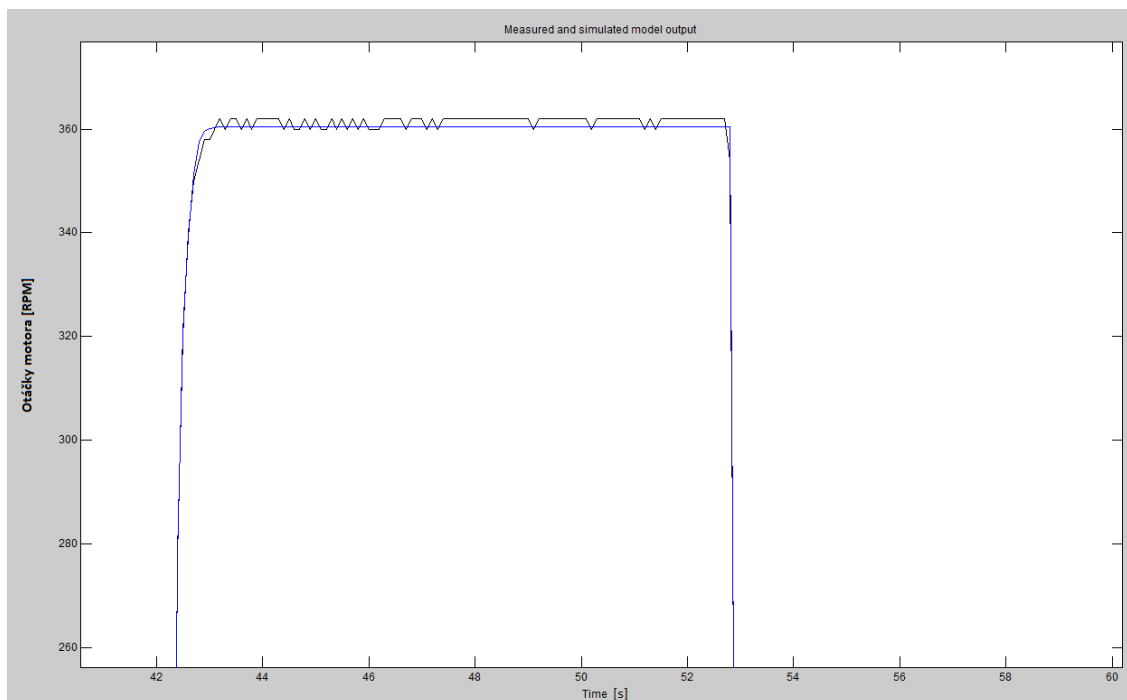
Identifikácia motoru z nameraných dát bol vyriešený s toolboxom v Matlabu. Namerané hodnoty boli načítané do programu Identification toolbox. V toolboxu môžeme namerané hodnoty graficky spracovať, filtrovať a vyhodnocovať.



15. Obrázok: Identification toolbox v Matlabe



14. Obrázok: Nameraný a teoretický model systému. Modrá odozva je teoretická, čierná odozva je nameraná



16. Obrázok: Nameraný a teoretický model systému. Modrá odozva je teoretická, čierná odozva je nameraná

Identifikovaná rovnica prenosu robota:

$$Fz(z) = \frac{0.6394 z^{-1} - 0.1224 z^{-2} + 0.07549 z^{-3} - 0.002586 z^{-4}}{1 - 0.8918 z^{-1} + 0.4738 z^{-2} - 0.2071 z^{-3} + 0.04243 z^{-4}} \quad (1)$$

Z odozvy s dá vidieť že medzi teoretickým modelom a skutočným, nameraným modelom sú len zanedbateľné líšenie. Nábehové a zostupné hrany modelu sú rovnaké a u ustálenie tak isto sú len zanedbateľné líšenie.

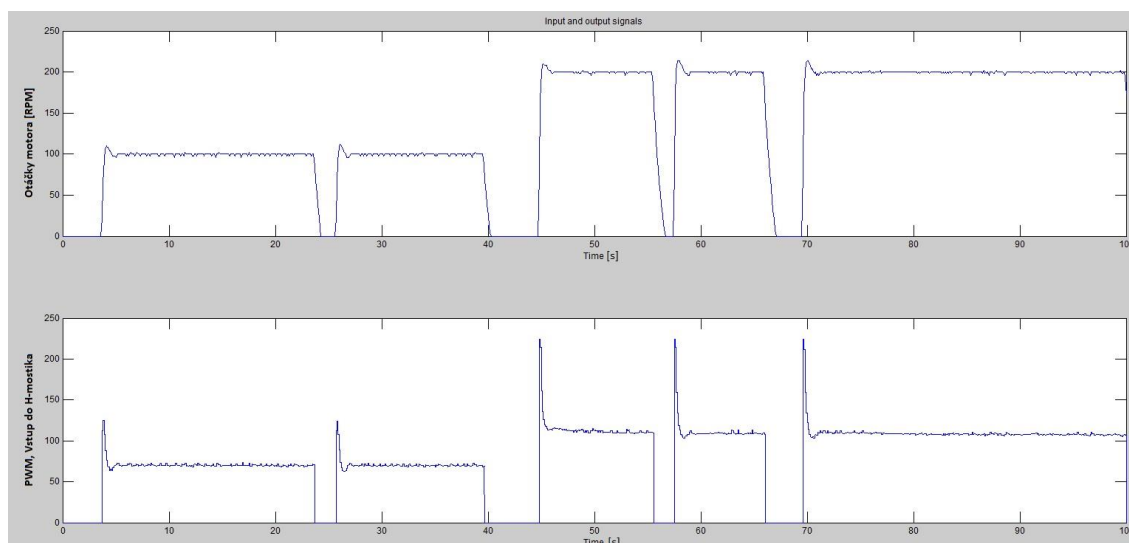
V arduine je používaná PID knižovňa [13] na spracovanie vypočítaných koeficientov. V knižovne sa dá nastaviť vzorkovaciu periódu, regulátorové konštanty a je aj vyriešená wind up jav.

Po pred-spracovní bol vykonaný návrh ARX modelu, a vykreslenie prechodovej charakteristiky.

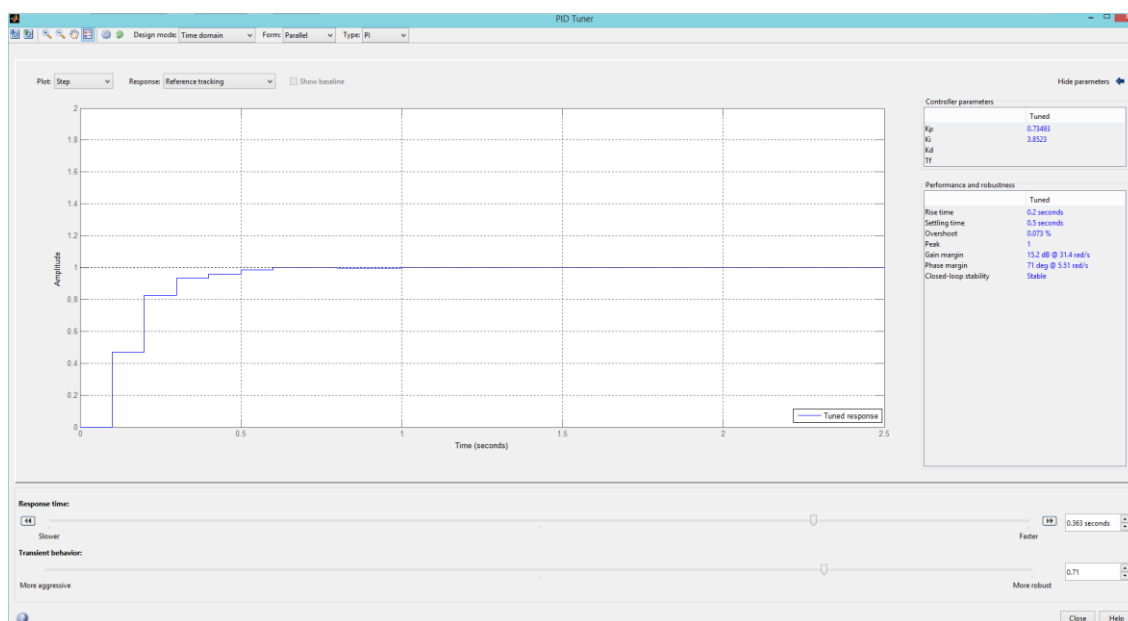
V praxi sa bežne používajú na reguláciu lineárnych sústav PID regulátor alebo PI. Na motory sa najčastejšie používajú PI regulátory. Nastavené a odskúšané regulátor na navrhovanom robote bola pre lepšiu voľbu PI regulátor, nakoľko má o 10 % rýchlejšiu nulovú ustálenú odchýlku pri menšom prekmite.

V Matlabe je PID toolbox na graficky ladenie regulátora. Máme možnosť nastaviť rýchlosť a charakter regulácie.

Výsledný regulátor má zosilnenie proporciálnej zložky  $K_p = 0.73493$ , integračnej zložky  $I = 3.8523$ . Doba ustálenia je 0.5 s. [31]

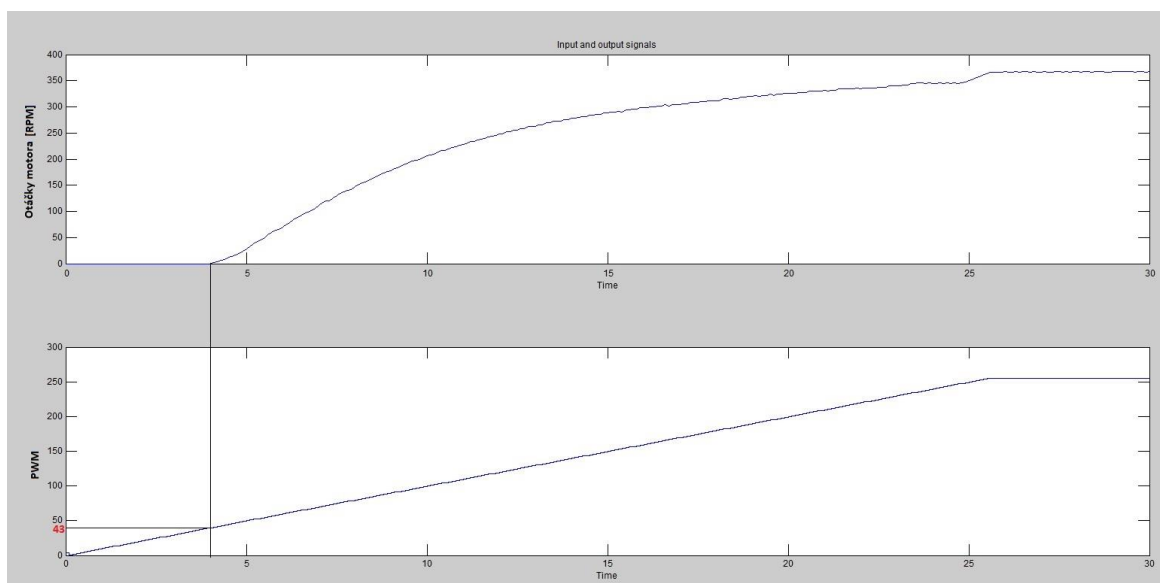


17. Obrázok: PI toolbox v Matlabe



18. Obrázok: Reakcie motora s PI regulátorom.

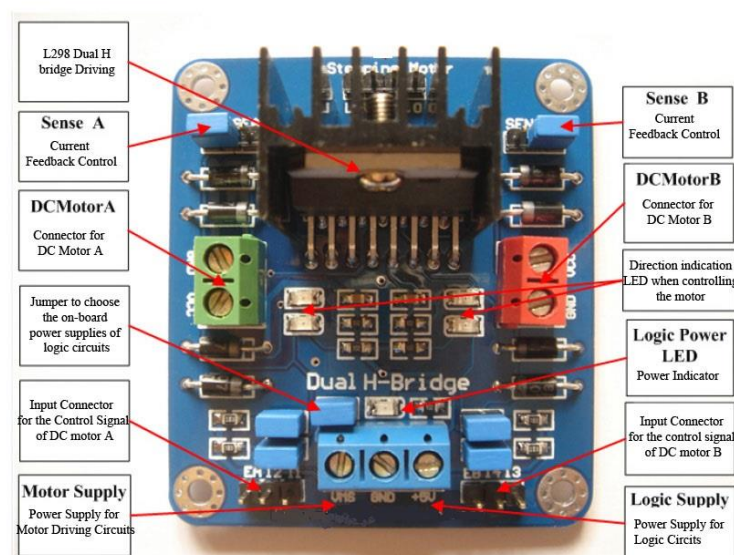
Na obrázku č.17 v hornom grafe sú znázornené otáčky motora a dolný vstupný signál. V hornom grafe sa dá vidieť ako reaguje regulátor. V prvej polovici bola nastavená požiadavka na 100 otáčok za minútu a v druhej polovici na 200 otáčok za minútu. Nelinearita motora bol identifikovaný s meraním. Meranie prebehlo tak, že PWM signál postupne zvýšilo podľa vpredem nadefinovanom čase, arduino meral otáčky motora a poslal pre matlabu. Matlab podľa namerané hodnoty vykrelsil graf. Podľa grafu motory reagujú na PWM signál od 40, to je úmerné na 15% “duty cycle”. Vyriešenie problematiky je dosť jednoduché, bol potrebný prepísať dolný mez výstupného signálu v knihovnu, ktorý je používané na regulátor.



19. Obrázok: Nelinearita motora

### 3.3.4 Ovládanie motora

Existuje mnoho spôsobov, ako vyriešiť problematiku smer otáčania motora. Najvhodnejší je h-mostik, nakoľko sa dajú veľmi jednoducho ovládať obidve smery motora. Na obrázku č. 19 shield s chipom L298. Najvhodnejší je zvoliť H-most, ako integrovaný obvod, kvôli veľkosti a nízkej cene.



20. Obrázok: Arduino L298 Shield [14]

L298 je Dual

Full-Bridge, ktorý

môže byť zaťažený 2 A na jeden kanál a maximálne 3 A. Dá sa riadiť s PWM signálom, má brzdivú funkciu a ochrana proti skratu a prepätia. Maximálne prevádzkové napätie je 50V. [46]

### 3.4 Operačný systém reálneho času

V rámci tohto projektu je používaná knihovňa [15] na RTOS. Knihovňa bola portovaná na Arduino z FreeRTOS a je open-source. Potreba RTOS je nevyhnutná, nakoľko v navrhovanom robote v jednej chvíli je potrebné sledovať alebo spracovávať viac požiadaviek. Príklad : keď sa počíta PI regulátor a treba aj sledovať akcelerometer alebo priat' dáta z počítača. Tieto skutočnosti nie sú možné bez operačného systému.

Používanie vlákna s knihovňou je veľmi jednoduché. Vytvorené vlákna sú nezávislé na druhom, a správajú sa ako hlavný program. Nie je potrebné ich vyvolať, ale je možné ich zastaviť a znova spustiť. Je potrebné nadefinovať čas pre vlákna a OS to skúsi vždy spustiť cyklicky, keď nastavený čas vypršal.

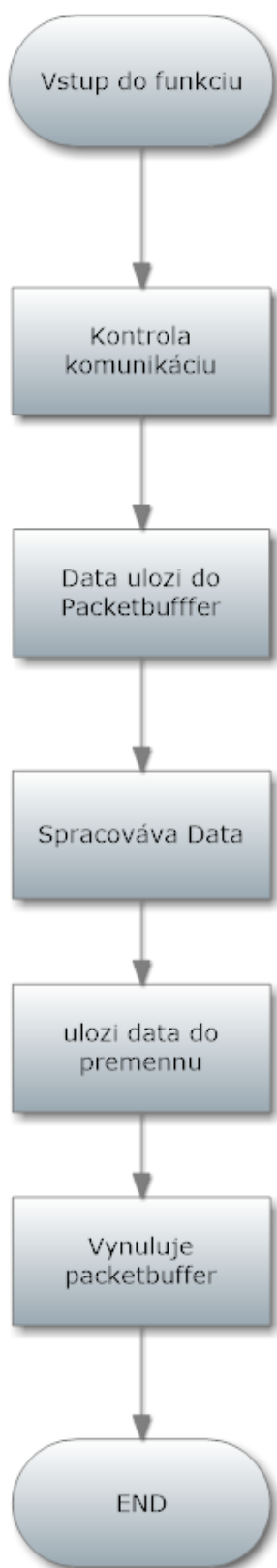
Najdôležitejšie vlákno je komunikácia. Každý 20 ms operačný systém spustí a funkcia uloží dáta čo dostala od klienta do takzvaného bufferu. Buffer používajú skoro v každej komunikácii, preto lebo keď naraz príde viac dát, a procesor nezvláda ich spracovanie môže dôjsť k strate niektorých dát. Buffer je v skutočnosti statická matica a dáta od klienta sú hneď uložené, a ostanú tam dovtedy, kým nebudú spracovávané a nebudú vynulované. Spracovávanie prebieha postupne, ako už bol vyššie popísané v kapitole komunikácia, pred a po každom príkaze je identifikačný znak a s príkazom substring() sa dajú identifikačné znaky vyhľadať v stringu. Dôsledkom je, keď informácie od klienta prídu poškodené alebo s obráteným poradím nebudú spracované. Vyhľadané informácie potom uloží do predom nadefinovanej premenné a s premennými ďalej pracujú ostatné vlákna. Netreba riešiť to či sa nemôže premenné vlákno prepísať alebo čítať informácie v jednom čase. Sú vytvorené vlákna ešte pre Regulátora, akcelerometra pre senzory a pre poslanie namerané hodnoty naspäť klienta.

Vlákno v ktorom je napísaný regulátor je volaný každých 50 ms. Vlákno každých 100 ms spraví výpočet rýchlosti motora, postup výpočtu je popísaný v kapitole 3.3.4 Ovládanie motora. Po výpočte vlákno vyvolá funkciu Compute() z knihovne a vypočíta regulátora pomocou predom zadanej koeficienty. Regulátor má zvlášť nastavenú vzorkovaciu periódu, a v funkciu Compute() skontroluje čas splnenia, aby nedošlo k tomu, že vlákno je volán viackrát než vzorkovacia perióda, vtedy aj viackrát počítal regulátora a tak by zmenil aj vzorkovacia perióda a diskretný regulátor mal by iné parametre. Takto vyriešená problematika zabezpečuje to že vzorkovacia perióda vždy bude 100 ms. Po výpočte nastaví výstup podľa toho, aká je požadovaná hodnota regulátora.

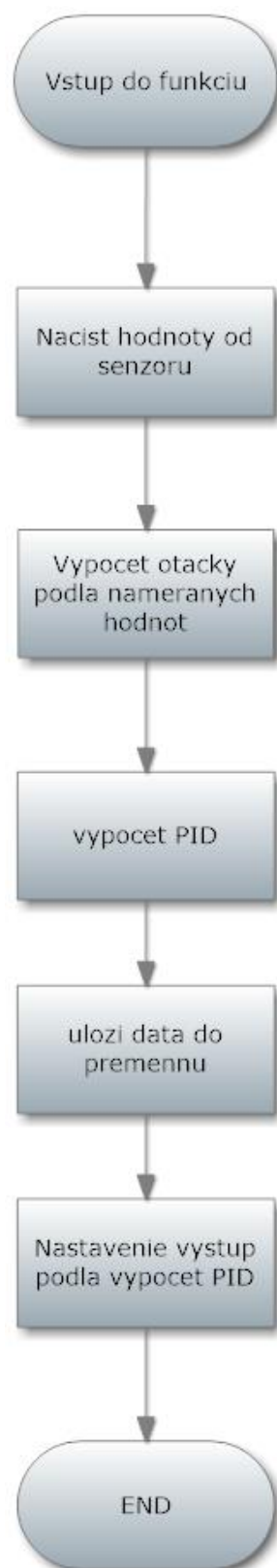
Vlákno na poslanie dát naspäť na server je vyvolaný každý 100 ms. Vlákno z nameranej hodnoty spraví stringovú maticu, tak že buffer, a postupne zvlášť pošle jednotlivé hodnoty.

Ostatné vlákna sú na riadenie a meranie senzorov, ktoré budú ďalej vysvetlené.

### 3.4.1 Tasky

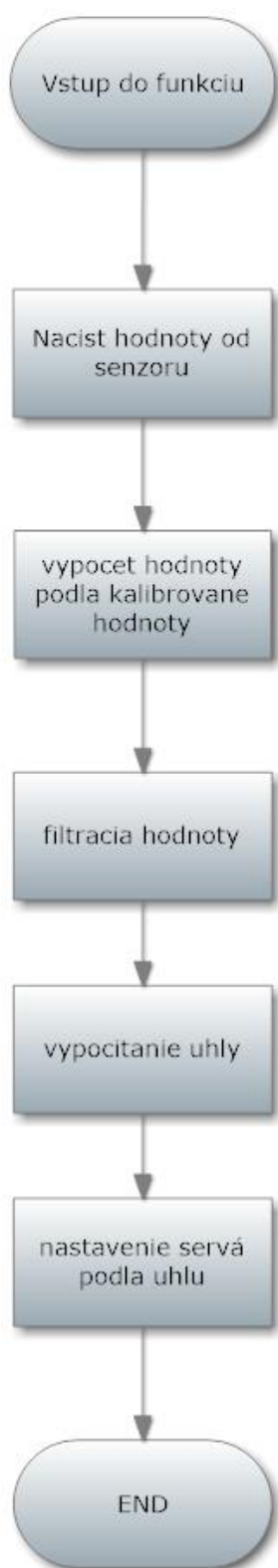


22. Obrázok: Diagram funkciu kumunikácia

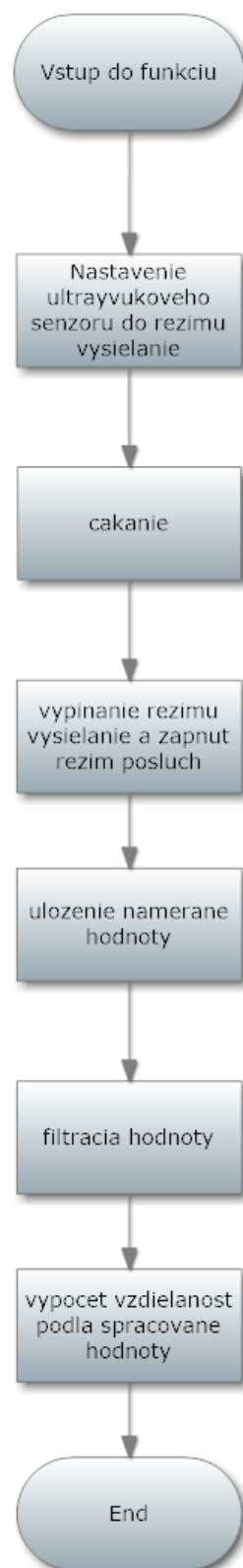


21. Obrázok: Diagram funkciu regulátor





24. Obrázok: Diagram funkciu Akcelerometer



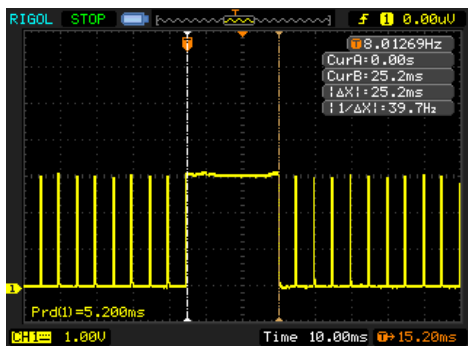
23. Obrázok: Diagram funkciu Ultrazvukového senzoru

### 3.4.2 Meranie vlákna

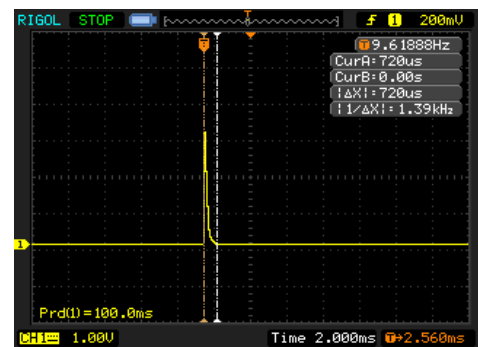
Dost' veľký problém je nastaviť vhodný čas cyklický volania vlákna. Najdôležitejší vlákno je komunikácia, ktorý je vyše podrobne popísaná. RTOS skúsi každý 5 ms spustiť vlákno, lebo v programe je funkcia na kontrolovania či prišiel už nový paket. Keď neprišiel tak vlákno skončí. Keď sú nové pakety vlákno udeľa spracovaní príkazu a to trvá 25,2 ms. Je to dost' veľa, ale preto lebo v tomto vlákne je navyiac riadenie manipulátoru a riadenie motora. Neprerušený časť vo vlákne trvá len 9  $\mu$ s.

Vlákno v ktorom je regulátor na motory, je volaný každý 100 ms. Vlákno vykoná procesor za 720  $\mu$ s.

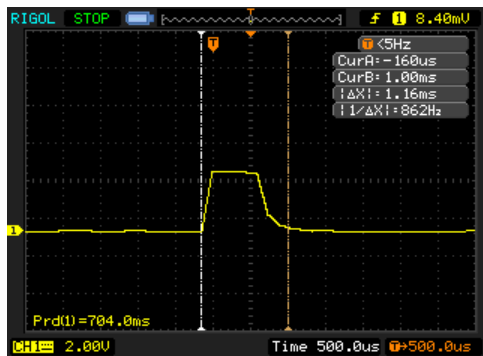
Ostatné vlákna nie sú tak dôležité, lebo pri je ich meškania len mešká meranie jednotlivých veličín, ale na hlavný funkčnosť robota nemá vplyv.



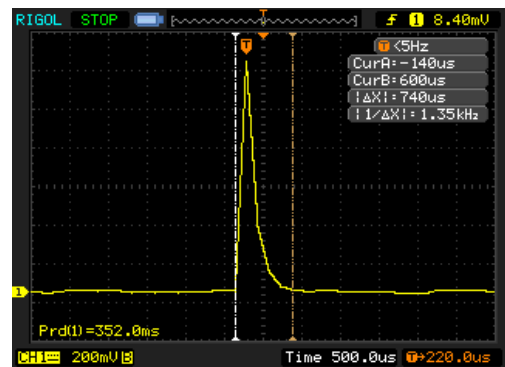
26. Obrázok: Vlákno komunikácia



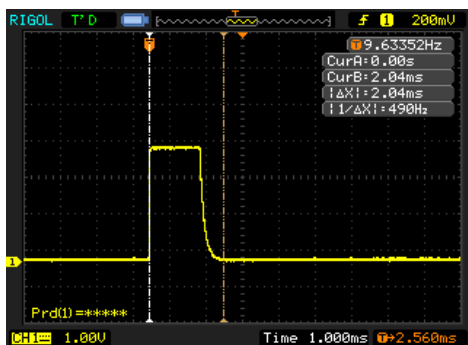
25. Obrázok: Vlákno Regulátora



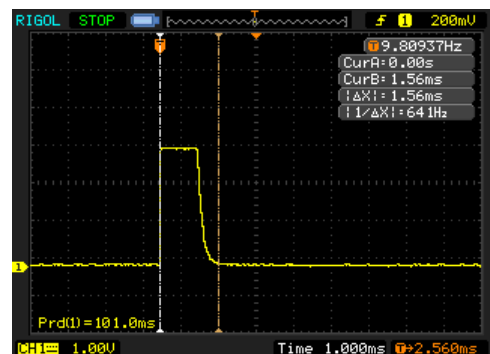
28. Obrázok: Vlákno svetelného senzora



27. Obrázok: Vlákno infra senzoru



30. Obrázok: Vlákno pre poslanie dáta



29. Obrázok: vlákno akcelerometra

## 3.5 Senzory

Senzory v robote slúžia pre ľahšie riadenia a na meranie rôznych fyzikálnych veličiny. V robote sú digitálne a analógové senzory. U digitálnych sú to také, ktoré komunikujú so zbernicou a sú také, ktoré len dajú na výstup logickú jednotku alebo nulu. Oba dva typy majú výhody aj nevýhody. Na komunikáciu so senzormi, ktoré majú výstup na zbernicu najčastejšie používajú knihovne, lebo skoro každý takýto senzor má viac módov. Sú to napríklad stand-by alebo meranie, dá sa u nich nastaviť aj rozlíšenie senzora alebo citlivosť. Všetky tieto nastavenia sú posielané v binárnom forme a s prednom napísanou knihovňou. Nastavenia sa dajú ľahko a jednoducho nastaviť. U analógových senzorov je potreba prevodu signálu zo spojitého na diskretný, aby sme mohli pracovať s hodnotami. V arduino DUE je 12-bitový A/D prevodník, s ktorým sa nameraná hodnota dá previesť na číslo od 0 až 4095.

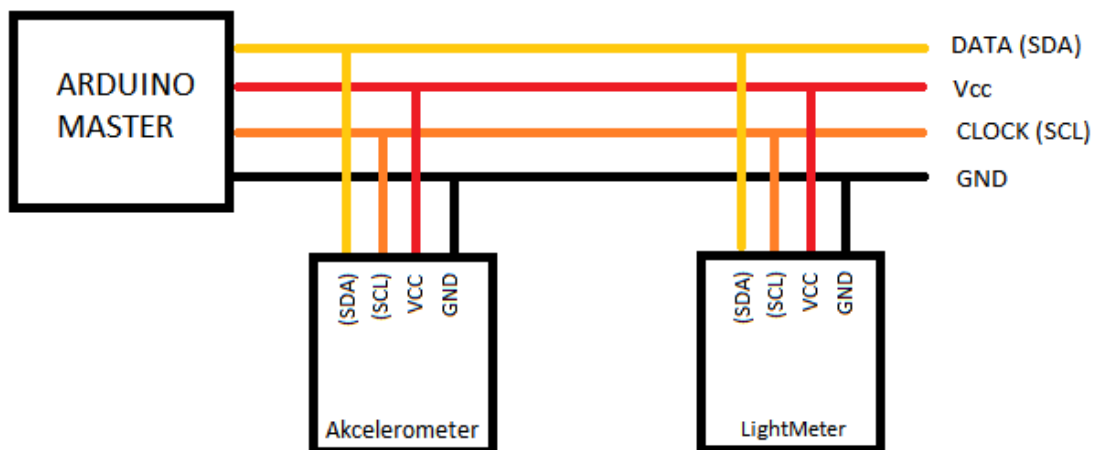
## 3.6 Digitálne senzory

### 3.6.1 Komunikácia I2C

I2C je dvojvodičová synchronná zbernica na prenos dát. Vyvinul spoločnosť Philips. Integrované obvody pripojíme z dvoma káblami: SCL (hodina), SDA (dáta).

Účastníci sa pripojujú s otvoreným drainom, alebo s otvorením kolektorom. Účastníkov môžeme adresovať, pričom adresa môže byť pevná alebo programovateľná.

V podstate jeden master a jeden alebo viac slave komunikujú medzi sebou. V systéme môže byť aj viac masterov. Vždy master vysiela hodinový signál na SCL. V minulosti rýchlosť presunu dát bol 100 kHz. Po verzii štandardu 3 bol dosiahnuteľný až 1 MHz. V 2012 vyvinuli verziu 4 štandardu s ktorým sa dá dosiahnuť rýchlosť až do 5 MHz, dnes sa všeobecne používajú štandardu 6, na ktorom vyvinuli úpravy ale maximálna dosiahnutá rýchlosť ostal 5 MHz. [16], [18]



31. Obrázok: Blokova schéma i2c senzorov

Prenos dát vždy začne master s tvorením štartovacej podmienky, následne pošle adresu, po adrese pošle jeden bit, ktorým signalizuje čítanie alebo písanie dát. Slave potvzuje prijatie s ACK a začne cyklus čítania alebo písania. Koniec prenosu dát signalizuje Master Stop podmienkou.

Takým spôsobom komunikuje v tejto navrhovanom robote akcelerometer a svetelný senzor.

### 3.6.1.1 Akcelerometer

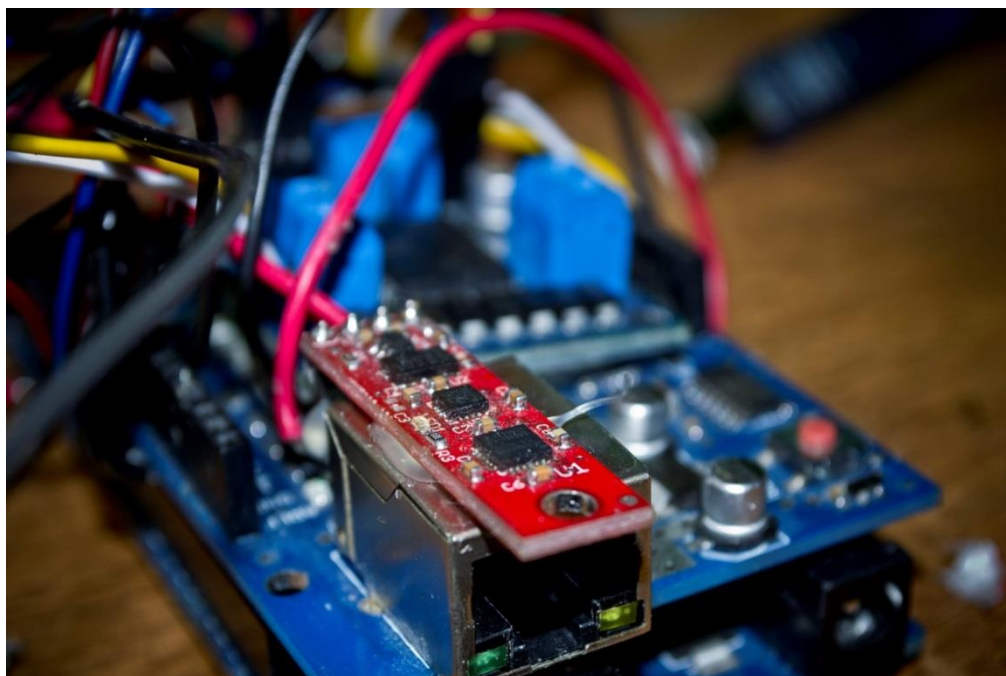
Akcelerometer typu ADXL345 [49] je integrovaný obvod na meranie zrýchlenia od firmy MEMS. Tento model je na doske 9DOF [50] s gyroskopom a magneto-metrom. ADXL345 meria zrýchlenie na troch osách v rozsahu  $\pm 16$  g. Má veľmi nízku spotrebu pri meraní zrýchlenia 40 uA a pri režime stand-by 0.1 uA.

V tomto projekte je používaný na sledovanie pohybu teľa robota a tak vytvoriť stabilizátor kamery. Podľa nameranej hodnoty sa vypočíta uhol natočenia teľa robota a podľa toho sa vygeneruje PWM signál. Generuje sa tak, aby servo otočilo kameru v protismere pohybu a tak vzniká stabilizácia kamery. Aby pohyby boli jemné je používaný kľzavé priemerovanie pre filtrácie hodnoty od akcelerometru. ADXL345 je napojené na I2C zbernicu. Meranie hodnôt prebieha cyklicky. Každých 100 ms vykoná cyklus na meranie hodnoty a nastavenie serva. [17], [30]

Výpočet stupňa sklonov je podľa vzorce [17] :

$$\text{pitch} = \text{atan2}(\text{accYval}, \text{accZval}) * \text{RAD\_TO\_DEG}; \quad (2)$$

$$\text{roll} = \text{atan2}(\text{accXval}, \text{accZval}) * \text{RAD\_TO\_DEG}; \quad (3)$$



32. Obrázok: 9DOF senzor s arduinom

### 3.6.1.2 Svetelný senzor

Svetelný senzor typu BH1750FVI je integrovaný obvod na meranie intenzity svetla. Komunikuje cez zbernicu i2c.

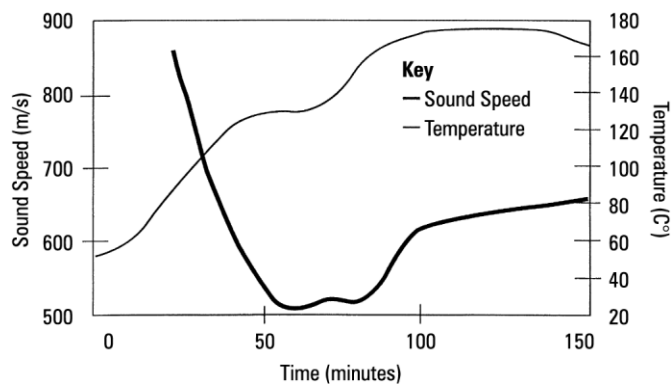
Senzor je schopný rozlíšiť zmenu svetla od 0.1 lx. Maximálne osvetlenie, ktoré dokážeme namerať je 65535 lx. Senzor je používaný na meranie svetla. Pokiaľ namerané hodnoty sú vyhodnotené a intenzita osvetlenia je menšia než predom nastavená hodnota, tak sa automaticky zapnú LED diódy vedľa kamery, tak aby aj pri menšom svetle bol kamerový obraz akceptovateľný. [19]



33. Obrázok: BH1750FVI senzor [53]

### 3.6.1.3 Ultrazvukový senzor

V projekte sú používané ultrazvukové senzory HC-SR04, na meranie vzdialeností, ktorý má rozsah od 2 cm do 4 m, pod uhlom 15°. Senzor má 4 vývody Trigger, Echo, Vcc, Gnd. Je používaný tak, že na Trigger je spustený vysoká úroveň signálu na 10  $\mu$ s a senzor automaticky vysiela osem 40 kHz ultrazvukový pulz. Výstup Echo sa nastaví na logickú jednotku dovedy, kým nevráti signál, ktorý je získaný odrazom od prekážky. V Arduine je funkcia pulseIn(), ktorá slúži na meranie impulzov a dá sa nadefinovať, pre ktorú úroveň má hodnotu sledovať. Funkcia vráti hodnotu v mikrosekundách. V tomto našom prípade je na definovaná tak, že bude sledovať logickú jednotku. Keď sa zadaný vstup prepne na logickú jednotku tak sa spustí časovanie a čaká sa dovedy, kým sa výstup neprepne na logickú nulu. Následne zastaví časovanie a výsledok je čas, ktorý je úmerný na dobu od vysielania ultrazvukového signálu a preto to ešte treba rozdeliť dvoma, aby výsledok bol len doba od vysielania do prekážok. Skutočný výsledok v cm je vypočítaný podľa rýchlosti ultrazvuku vo vzduchu. Funkcia spracováva od 10 ms do 3 s. To znamená že najmenší vzdialenosť čo sa dá takým spôsobom merať je podľa vzorca  $3.7 \cdot 1.7 \text{ mm}$ , to znamená že rozlíšenie je 0.17 cm. V suchom vzduchu pri teplote 24 °C je šírenie ultrazvuku je 340 m/s, ktorý je 0.034 cm/ $\mu$ s. Nedá sa zanedbať, že pri inej teplote je rýchlosť šírenia ultrazvukového signálu zmení, a preto senzor môže byť nepresný pri niektorých situáciách. Často sa kombinujú ultrazvukový senzor ešte teplomerom a vlhkomerom. [20], [21]



34. Obrázok: Rýchlosť šírenia zvuku pri zmene teploty[44]

Približná rýchlosť zvuku v suchom vzduchu sa dá vypočítať podľa vzorca:

$$c = 331.5 + 0.6 * [\text{teplota vzduchu v Celzia}] \quad \text{m / s} \quad (4)$$

Výsledok rýchlosti zvuku ešte treba vynásobiť  $1 \cdot 10^{-4}$  a tak je rýchlosť bude v cm /  $\mu\text{s}$ .

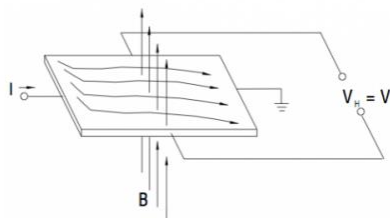
$$\text{Vzdialenosť} = \left( \frac{\text{nameraný čas}}{2} \right) * ((331.5 + 0.6 * [\text{teplota vzduchu v Celzia}]) * 1 * 10^{-4})$$

[21] (5)

### 3.6.1.4 Hallový senzory

Tieto senzory sa používajú kvôli Hallova javu. Keď Hallovou sondou tečie prúd a je umiestnená do magnetického poľa, tak že jeho siločiaru sú kolmé k polovodičovej dosičke a teda aj k smeru pretekajúceho prúdu, sú elektróny, ktoré dosičkou prechádzajú, vplyvom magnetického poľa vychýľované. Sila, ktorá elektróny vychýľuje z priameho smeru vždy k jednej strane dosičky, sa nazýva Lorenzova sila [50].

Takým istým princípom funguje aj snímač otáčok na motore v robote. Motor otáča kruhový magnet, ktorý je napojený na hriadeľ motora a vedľa magnetu je hallov senzor. Magnet nie je úplne symetrický, má jedno dorezanie, ktoré slúži na prerušenie magnetického poľa. Na výstupe senzora je kmitavý signál, z ktorého sa dajú vypočítať otáčky motora.



36. Obrázok: Hallova jev [26]



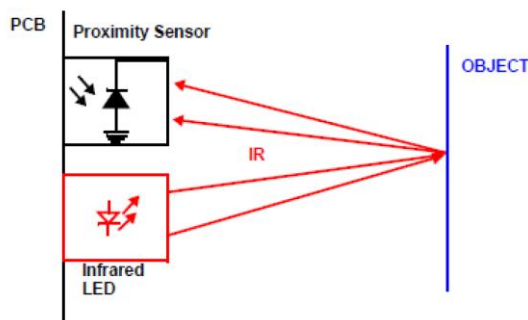
35. Obrázok: Motor s Hallov senzorom [43]

## 3.7 Analogové sensory

### 3.7.1.1 IRsensor

Infra senzory sú používané na meranie blízkosti objektu bez fyzického kontaktu. V tomto projekte je spolu používaní s ultrazvukovým senzorom. Preto bol zvolený infračervený senzor lebo tie dve typy senzorov, ktoré sú používané na meranie blízkosti objektu nemajú žiadne rušujúce vplyvy na druhé. Infračervený senzor využije tú istú jav ako ultrazvukový senzor, ale pomocou infračerveného svetla. Skladá sa z infra LED-u a infra prijímaču. Infra LED vysiela infračervené svetlo, ktorý odráža od objektu a vráti sa do prijímača Ir. senzoru. Tento princíp sa volá Proximity.

V tomto projekte je používaný infra senzor od firmy Sharp 2Y0A21 F9Z, s rozsahom merania od 10 do 80 cm. V projekte je obmedzená rozsah do 30 cm. Podľa skúsenosti senzor je dosť presný i aj pri menšom vzdialenosti než 10 cm. Výstup senzoru je napätie, ktorý je úmerne na vzdialenosť. Preto tento senzor je napojený na analógový vstup arduina. Pomocou 12 bitovú A/D prevodníku, signál je vzorkovaný a tak merané napätie je prepočítaná na vzdialenosť. [52]. [23]



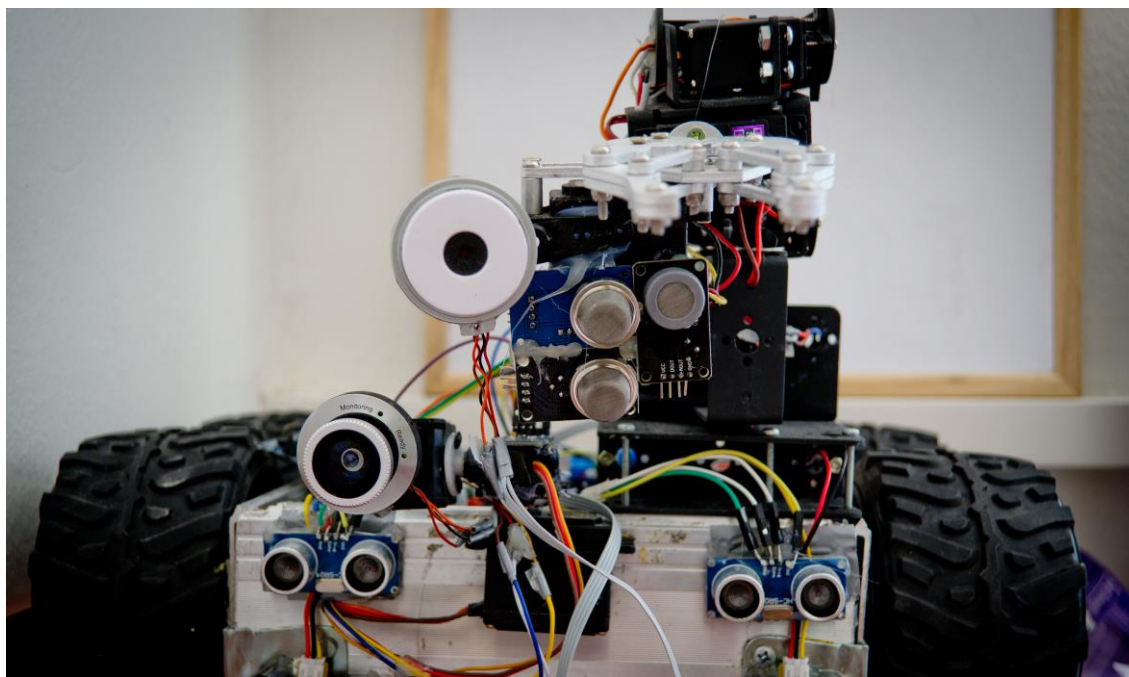
37. Obrázok: Princíp proximity s Ir. senzorom [24]

## 3.8 Kamera

V robote je zabudovaná IP kamera Edimax IC-1500. Kamera má maximálne rozlíšenie 640x 480 (VGA) a pri tomto rozlíšení max 30 fps. Senzor je farebný CMOS . Fókus je manuálny F=1.8. Obráz kamery je integrovaný do klient programu s CGI príkazom. Firma Edimax zverejnil CGI príkazy na kameru, a tak bolo možné implementovať jeho stream do programu. Podľa dokumentu priami stream kamery po prihlásení heslom sa dá dosiahnuť na adrese [HTTP://camera\\_ip/mjpg/video.mjpg](http://camera_ip/mjpg/video.mjpg). Celá kamera je rozobratá na dve časti : základná doska je v tele robota a optika je s káblom vyvedená na telo robota. [25]



Základná doska tejto kamery je kompatibilná s iným optikou, preto na robotické rameno je namontovaná ešte jedna optika a obraz je pripínateľný z jednej optiky na druhú. Kamera komunikuje so základnou doskou cez 4 káble. Pomocou 4 prepínacie relemý s klientskym softwarom sa dá prepínať. Optika takých starších a nekvalitných kamier majú často úzky uhol pohľadu, preto je na optiku namontovaný fisheye objektív. Fisheye objektív je ultra širokouhlý objektív, ktoré vytvorí vizuálne skreslenie, určené na vytvorenie širokého panoramatického obrazu



38. Obrázok: Kamery na robote

### 3.9 Robotické rameno

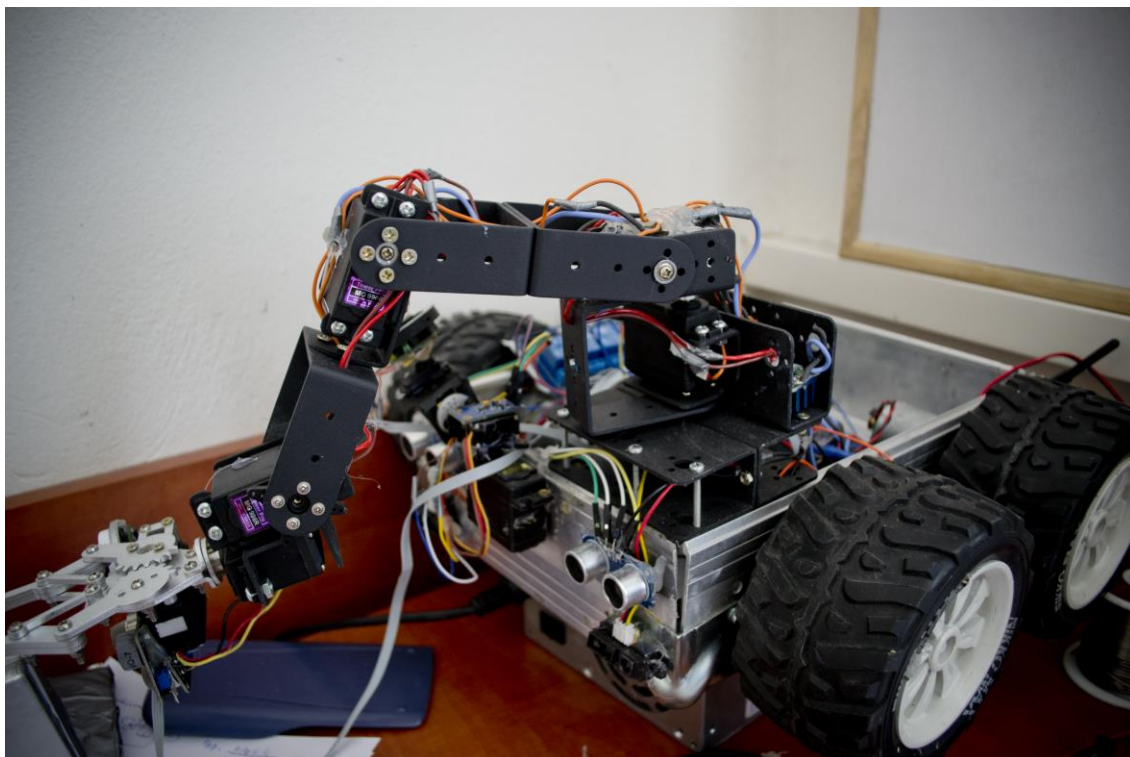
Robotické rameno sa skladá zo 6 servo pohonov a z držiaku, ktoré držia servá. Na konci ramena je namontovaný klips a nato je pridaná druhá optika IP kamery. Servo pohony sú riadené s PWM signálom, ktoré vygeneruje Arduino. Servá sú digitálne, typu MG996R, ktoré majú kovové pastorky. Operačné napätie majú od 4,7 V do 7,2 V, pri 5 V majú v vyvinú silu 0.98 Nm a rýchlosť  $60^\circ / 0,20\text{sec}$ . [53]

Riadenie celého ramena je zabezpečené pomocou klientskeho softwaru. Software sleduje pohybovanie myšom alebo touchpadom a kopíruje pohyby. Podľa pohybu prepočíta uhly a nastaví servá na požadovanú pozíciu. Každé servo má obmedzenie pohybu od  $0^\circ$  do  $180^\circ$ . Pri pohybovaní myši smerom dopredu sa aj rameno pohybuje, tak že natočenie niektorého serva sa zvýši a pre niektoré sa zníži. Keď je potrebné zvlášť pohybovať s jednotlivými časťami ramena, tak sa dá príkaz skombinovať za pomoci tlačidla a kolieskom na myši. Pri pohybovaní s ramenom nám pomáha aj druhá kamera, pomocou ktorej sa dá sledovať obraz v perspektíve.



### 3.10 Napájanie

Potrebný napájací napätie poskytujú DC/DC konvertory, ktoré sú nastaviteľný v rozsahu od 0,1 V až do napätia ktorý je na vstupe s úbytkom 0.5 V. Maximálne vstupné napätie je 28 V. Maximálni zaťažovací prúd je 3 A, ale podľa skúsenosti s chladením môže to byť oveľa viac. Potrebné napätie môžeme nastaviť s trimermi. V robote je 6 kusov DC/DC konvertoru, ktoré sú napojené sériovo, lebo, keď servá boli na jednom konvertoru s senzormi vtedy vzniklo rušenie. Kvôli rušenia jednotlivé hlavné časti robota sú na zvláštnom konvertoru.



39. Obrázok: Robotické rameno

	Prúd [I]	Napätie [V]
Router	1.0 A	10.7 V
Arduino DUE	0.5 A	5.0 V
Kamera	1.0 A	10.7 V
Senzory	0.7 A	5.0V
Serva na pohybovanie ramena	2.0 A	5.2 V
Serva na pohybovanie kameru	0.5 A	5.2 V

2. Tabuľka: Spotreba prúdu a napätia komponentu

Napájanie poskytuje akumulátor, ktorý je typu Li-Po. Parametry akumulátoru sú: 11,1 V, 5000 mAh a 25 C. V tomto projekte je používaný špeciálny akumulátor , ktoré používajú v modelárstve. Li-Po akumulátory majú približne 1-2 C, ale špeciálne modelárske akumulátory majú 20 – 50 C. Tento model má 25 C to znamená že jeho

vybíjací prúd je 125 A. Li-Po akumulátory sú veľmi kazlivé na ohľadu vybíjania. Nabitý Li-Po článok má 4,23 V, a keď jeho napätie klesne pod 2,7 V akumulátor môže byť zničený. Preto do akumulátory namontujú elektrický obvod pre chránenia akumulátora proti prebitiu, prílišnému vybitia a proti prehrievania. Bohužiaľ do lacnejšie akumulátory nie sú chránené takým spôsobom. Bol by potrebné to externe vyriešiť. [22]



41. Obrázok: Li - Po akumulátor [55]



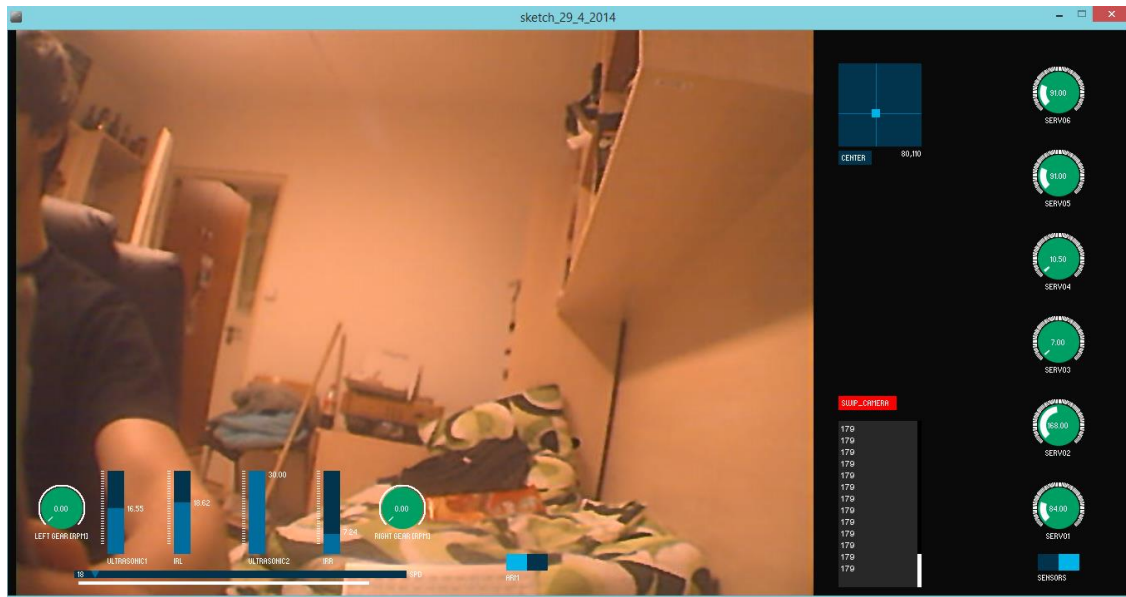
40. Obrázok: DC / DC konvertor [54]

### 3.11PC aplikácia

Aplikácia na riadenie robota bol písaný v Processingu. Processing je open source programovací jazyk a integrovaný vývojový prostredie. Jazyk je založený na Java, ale používa sa jednodušený syntax a grafický programovací model. [58]

Processing má to výhodu že uľahčí prácu s arduinom, lebo snažia sa kombinovať arduino a processing. Tak isto ako u arduino sú dostupné mnoho tutoriali a externých knižovny.

V tomto projekte program, ktorý je napísané v processingu je používaný na riadenia robota, riadenie robotického ramena, na sledovanie kamerového obrazu a na vyzualizáciu nameraných hodnoty od senzorov. Z programu sa dá aktivovať alebo deaktivovať manipulátor, riadiť. Dá sa zapnúť alebo vypnúť senzory, prepínať kamerový obraz, hýbať s kamerou, nastaviť rýchlosť otáčania motora. ,[38], [39], [40]



42. Obrázok: Počítačová aplikácia na riadenie

### 3.12Shrnutie

Výsledok práce je funkčný robot, s riadiacim systémom, s vyriešeným komunikáciou medzi mikrokontrolérom a počítačom cez Wi – Fi sieť.

V projekte je používaný senzor na meranie zrýchlenia, a tým je vytvorený stabilizátor kamery. Sú používané ultrazvukové, infračervené proximity senzory na meranie vzdialenosti robota od prekážok. Na robote bol zostavený funkčný robotické rameno. V robote je zabudovaná IP kamera, ktorý má dve optiky a pomocou programu sa dá sledovať obraz a vybrať optiku a tak perspektívu.

Ďalej je napísaný testovací program do počítača na: ovládanie z diaľku robota s klávesnicou, na snímanie obrazu z kamery, na pohybovanie kameru pomocou klávesnici alebo s myšom, na pohybovanie manipulátoru s myšom. Program vizualizuje namerané hodnoty.

## 4 ZÁVER

V tomto projekte bol navrhovaný a vyrobený tzv. teleprezenčný robot, ktorý má za účelom merať fyzikálne veličiny v okolí robota.

V prvej časti projektu sa zaoberáme s teoretickým vysvetlením jednotlivých častí robota. Bolo to hlavne pojímanie teleprezencie, rozoberaný embedded system Arduino a jednotlivé typy Arduina. Následne boli posudzované výhody i nevýhody použitých komponentov. Nami zvolený arduino používa ARM Cortex M-3 processor, preto sú v teoretickej časti zhrnuté hlavné parametre a architektúra procesora. V projekte je používaný operačný systém reálneho času, preto sú v teórii vysvetlené typy OS. Sú to hlavne princípy činností a ich jednotlivé parametre.

V druhej, praktickej časti projektu bol riešený obojsmerný klient – server, komunikácia s UDP protokolom. Ďalej bola predstavená celková architektúra robota s jeho výhodami a nevýhodami. Jednotlivé časti navrhovania regulátora ako : meranie otáčok, posielanie hodnôt do počítača, identifikácia alebo nastavenie regulátora sú postupne popísané v jednotlivých častiach projektu. Na výpočet regulátora a na identifikáciu motora bol používaný výpočtový program Matlab. V Matlabe bol využívaný Identification toolbox a PID toolbox.

V ďalšej časti projektu je popísané realizovanie nastavenia vlákna pomocou operačného systému reálneho času.

V poslednej časti sa práca zaoberá vyriešením komunikácie digitálnych, analógových senzorov a spracovanie nameraných hodnôt od senzorov. Bolo riešené riadenie servo pohonov pre robotické rameno a pre pohyb s kamerou. Zo strany počítača bol realizovaný program pre riadenie robota, riadenie robotického ramena, monitorovanie meraných hodnôt a sledovanie obrazu z kamery. Namerané dáta od senzorov sú poslané pre klientsky program pre vizualizáciu a pre presnejšie riadenia robota.

# Literatura

- [1] Sina.com.cn [online]. 2008 [cit. 2014-05-24]. Dostupné z: <http://mil.news.sina.com.cn/2007-09-30/0847466410.html>
- [2] DEL FRESNO EN, David. UGV ACTUAL [online]. 2013 [cit. 2014-05-24]. Dostupné z: <http://ugvactual.blogspot.cz/2013/10/ugv-models-face-off-over-firepower-load.html>
- [3] LOMBARD, Matthew. Giantscreencinema.com: Using Telepresence to Communicate Science in Giant Screen Cinema [online]. 2011 [cit. 2014-05-24]. Dostupné z: <http://www.giantscreencinema.com/Portals/0/LombardPresentationRev.pdf>
- [4] Store.arduino.cc: Arduino Boards [online]. 2014 [cit. 2014-05-24]. Dostupné z: [http://store.arduino.cc/index.php?main\\_page=index&cPath=11](http://store.arduino.cc/index.php?main_page=index&cPath=11)
- [5] Arduino [online]. 2014 [cit. 2014-05-24]. Dostupné z: <http://arduino.cc/>
- [6] Úvod do architektury Cortex-M3 - díl. 1. Pandatron.cz [online]. 10.02.2010, č. 1 [cit. 2014-01-13]. Dostupné z: [http://pandatron.cz/?1252&uvod\\_do\\_architektury\\_cortex-m3\\_-\\_díl.\\_1](http://pandatron.cz/?1252&uvod_do_architektury_cortex-m3_-_díl._1)
- [7] Praktické zkušenosti s procesory ARM Cortex M3. Hw.cz [online]. 7.4.2009, č. 1 [cit. 2014-01-13]. Dostupné z: <http://www.hw.cz/soucastky/embedded-systemy/mcu/prakticke-zkusenosti-s-procesory-arm-cortex-m3.html>
- [8] OTAVA, Lukáš. Firmware pro robotické vozítko [online]. Brno, 2013 [cit. 2014-05-24]. 69 s. ISBN 80-214-2610-1. Firmware pro robotické vozítko. Dostupné z: [https://www.vutbr.cz/www\\_base/zav\\_prace\\_soubor\\_verejne.php?file\\_id=64104](https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=64104). Diplomová práce. Vysoké učení technické v Brně. Vedoucí práce Ing. PAVEL KUČERA, Ph.D.
- [9] ZEŽULKA, František. Prostředky průmyslové automatizace [online]. 1. vyd. Brno: VUTUM, 2004, 176 s. [cit. 2014-05-24]. ISBN 80-214-2610-1. Dostupné z: [https://www.vutbr.cz/www\\_base/priloha.php?dpid=21239](https://www.vutbr.cz/www_base/priloha.php?dpid=21239)
- [10] Wikipedia: Klient-server [online]. 2014 [cit. 2014-05-24]. Dostupné z: <http://sk.wikipedia.org/wiki/Klient-server>
- [11] KAS. Forum.arduino.cc: DC motor control with PID [online]. 2010 [cit. 2014-05-24]. Dostupné z: <http://forum.arduino.cc/index.php?PHPSESSID=5hq70cogvukg5s328nvl5b1r1&topic=8652.0>

- [12] Arduino.cc: PWM [online]. 2014 [cit. 2014-05-24]. Dostupné z: <http://arduino.cc/en/Tutorial/PWM>
- [13] Arduino PID Library. BRETT, Beauregard. Playground.arduino.cc [online]. 2013 [cit. 2014-01-16]. Dostupné z: <http://playground.arduino.cc/Code/PIDLibrary>
- [14] Cutedigi.com: LinkSprite Arduino Robotics Kit [online]. 2014 [cit. 2014-05-24]. Dostupné z: <http://www.cutedigi.com/robotics/linksprite-arduino-robotics-kit.html>
- [15] ArduinoThread. SEIDEL, Ivan. GitHub.com [online]. 1. vyd. 11.05.2013 [cit. 2014-01-16]. Dostupné z: <https://github.com/ivanseidel/ArduinoThread>
- [16] Embedded-lab.com: INTER-INTEGRATED CIRCUIT (I2C) COMMUNICATION [online]. 2011 [cit. 2014-05-24]. Dostupné z: <http://embedded-lab.com/blog/?p=2583>
- [17] Forum.arduino.cc: Guide to gyro and accelerometer with Arduino including Kalman filtering [online]. 2011 [cit. 2014-05-24]. Dostupné z: <http://forum.arduino.cc/index.php/topic,58048.msg417140.html#msg417140>
- [18] I2cdevlib. JROWBERG. GitHub.com [online]. 2013 [cit. 2014-01-16]. Dostupné z: <https://github.com/jrowberg/i2cdevlib>
- [19] Elechouse.com: BH1750FVI Digital Light Sensor Module [online]. 2011 [cit. 2014-05-24]. Dostupné z: [http://www.elechouse.com/elechouse/index.php?main\\_page=product\\_info&cPath=&products\\_id=2159](http://www.elechouse.com/elechouse/index.php?main_page=product_info&cPath=&products_id=2159)
- [20] Arduino Basic: HC-SR04 Ultrasonic Sensor [online]. 2012 [cit. 2014-05-24]. Dostupné z: <http://arduinobasics.blogspot.cz/2012/11/arduino-basics-hc-sr04-ultrasonic-sensor.html>
- [21] Trollmaker.com: Arduino and HC-SR04 ultrasonic sensor [online]. 2011 [cit. 2014-05-24]. Dostupné z: <http://trollmaker.com/article3/arduino-and-hc-sr04-ultrasonic-sensor>
- [22] SALT, John. Rchelicopterfun.com: Understanding RC LiPo Batteries [online]. 2014 [cit. 2014-05-24]. Dostupné z: <http://www.rchelicopterfun.com/rc-lipo-batteries.html>
- [23] Luckylarry.co.uk: Arduino – Using a Sharp IR Sensor for Distance Calculation [online]. 2009 [cit. 2014-05-24]. Dostupné z: <http://luckylarry.co.uk/arduino-projects/arduino-using-a-sharp-ir-sensor-for-distance-calculation/>
- [24] LUO, Yong a Tamara SCHMITZ. Eetimes.com: Infrared proximity sensing: Building blocks, mechanical considerations, & design trade-offs [online]. 2009 [cit. 2014-05-24]. Dostupné z: [http://www.eetimes.com/document.asp?doc\\_id=1272536](http://www.eetimes.com/document.asp?doc_id=1272536)

- [25] Edimax: Network Cameras IC-1500 [online]. 2008, 2013 [cit. 2014-05-24]. Dostupné z: [http://www.edimax.com/au/produce\\_detail.php?pd\\_id=54&pl1\\_id=8&pl2\\_id=35](http://www.edimax.com/au/produce_detail.php?pd_id=54&pl1_id=8&pl2_id=35)
- [26] EEWeb: Hall Effect Sensors [online]. 2011 [cit. 2014-05-24]. Dostupné z: <http://www.eeweb.com/electronics-quiz/hall-effect-sensors/>
- [27] Wikipedia: Real-time operating system [online]. 2014, 14.4.2014 [cit. 2014-05-24]. Dostupné z: [http://en.wikipedia.org/wiki/Real-time\\_operating\\_system](http://en.wikipedia.org/wiki/Real-time_operating_system)
- [28] National Instruments: What is a Real-Time Operating System (RTOS)? [online]. 2013 [cit. 2014-05-24]. Dostupné z: <http://www.ni.com/white-paper/3938/en/>
- [29] The C Continuum: Arduino IMU: Pitch & Roll from an Accelerometer [online]. 2012 [cit. 2014-05-24]. Dostupné z: <http://thecontinuum.com/2012/09/24/arduino-imu-pitch-roll-from-accelerometer/>
- [30] Bayesianadventures.wordpress.com/: Gyroscopes, Accelerometers and the Complementary Filter [online]. 2013 [cit. 2014-05-24]. Dostupné z: <http://bayesianadventures.wordpress.com/2013/10/20/gyroscopes-accelerometers-and-the-complementary-filter/>
- [31] Mathworks.com: pidtool [online]. 2014 [cit. 2014-05-24]. Dostupné z: <http://www.mathworks.com/help/control/ref/pidtool.html>
- [32] I2cdevlib.com: I2C [online]. 2010 [cit. 2014-05-24]. Dostupné z: <http://www.i2cdevlib.com/>
- [33] Upgrayd.blogspot.cz: PID Velocity DC Motor Controller [online]. 2011 [cit. 2014-05-24]. Dostupné z: <http://upgrayd.blogspot.cz/p/pid-velocity-motor-controller.html>
- [34] Bubbleweb.eu: Are We Entering The Post-PC Era? [online]. 2012 [cit. 2014-05-24]. Dostupné z: <http://www.bubbleweb.eu/are-we-entering-the-post-pc-era/>
- [35] COMNES, Bret a A. LA ROSA. Pdx. Pdx.edu: Arduino PID Example Lab [online]. 2013, č. 1, s. 7 [cit. 2014-05-24]. Dostupné z: [http://www.pdx.edu/nanogroup/sites/www.pdx.edu.nanogroup/files/2013\\_Arduino%20PID%20Lab\\_0.pdf](http://www.pdx.edu/nanogroup/sites/www.pdx.edu.nanogroup/files/2013_Arduino%20PID%20Lab_0.pdf)
- [36] Mathworks.com: Arduino Support from Simulink [online]. 2013 [cit. 2014-05-24]. Dostupné z: <http://citace.lib.vutbr.cz/dokument/eQuJ1PUXTtuUXVFI>
- [37] Mathworks.com: Motor Control with Arduino: A Case Study in Data-Driven Modeling and Control Design [online]. 2013 [cit. 2014-05-24]. Dostupné z: <http://www.mathworks.com/company/newsletters/articles/motor-control-with-arduino-a-case-study-in-data-driven-modeling-and-control-design.html>

- [38] FRY, Ben a Casey REAS. Processing.org/: Capture [online]. 2014 [cit. 2014-05-24]. Dostupné z:<http://processing.org/reference/libraries/video/Capture.html>
- [39] Learning.codasign.com: Mouse Events in Processing [online]. 2012 [cit. 2014-05-24]. Dostupné z: [http://learning.codasign.com/index.php?title=Mouse\\_Events\\_in\\_Processing](http://learning.codasign.com/index.php?title=Mouse_Events_in_Processing)
- [40] Wiki.processing.org/: Processing for Android [online]. 2014 [cit. 2014-05-24]. Dostupné z:<http://wiki.processing.org/w/Android>
- [41] BEAUREGARD, Brett. Brettbeauregard.com: Improving the Beginner's PID: Reset Windup [online]. 2011 [cit. 2014-05-24]. Dostupné z: <http://brettbeauregard.com/blog/2011/04/improving-the-beginner%E2%80%99s-pid-reset-windup/>
- [42] I.ebayimg.com/ [online]. 2014 [cit. 2014-05-24]. Dostupné z:[http://i.ebayimg.com/00/s/NTAwWDUwMA==/z/UDUAAOxyOlhSzhrO/\\$\\_12.JPG](http://i.ebayimg.com/00/s/NTAwWDUwMA==/z/UDUAAOxyOlhSzhrO/$_12.JPG)
- [43] Pololu.com: 30:1 Metal Gearmotor 37Dx52L mm with 64 CPR Encoder [online]. 2011 [cit. 2014-05-24]. Dostupné z: <http://www.pololu.com/product/1443/>
- [44] Sensor review: international journal of sensing for industry [online]. 3. vyd. Kempston: I.F.S.Publications, 1999 [cit. 2014-05-24]. ISBN 0260-2288. Dostupné z:<http://www.emeraldinsight.com/journals.htm?articleid=1454652>
- [45] Elechouse.com: WiShield 3.0, Arduino WiFi [online]. 2012 [cit. 2014-05-24]. Dostupné z:[http://www.elechouse.com/elechouse/index.php?main\\_page=product\\_info&cPath=90\\_186&products\\_id=2165](http://www.elechouse.com/elechouse/index.php?main_page=product_info&cPath=90_186&products_id=2165)
- [46] Sparkfun.com: DUAL FULL-BRIDGE DRIVER L298 [online]. 2000 [cit. 2014-05-24]. Dostupné z:[https://www.sparkfun.com/datasheets/Robotics/L298\\_H\\_Bridge.pdf](https://www.sparkfun.com/datasheets/Robotics/L298_H_Bridge.pdf)
- [47] Sparkfun.com: Digital Accelerometer ADXL345 [online]. 2009 [cit. 2014-05-24]. Dostupné z:<https://www.sparkfun.com/datasheets/Sensors/Accelerometer/ADXL345.pdf>
- [48] Sparkfun.com: 9 Degrees of Freedom [online]. 2012 [cit. 2014-05-24]. Dostupné z:<https://www.sparkfun.com/products/10736>
- [49] Micropik.com: Ultrasonic Ranging Module HC - SR04 [online]. 2009 [cit. 2014-05-24]. Dostupné z:<http://www.micropik.com/PDF/HCSR04.pdf>
- [50] NETÁHLO, Tomáš. Cyklopočítač s měřičem frekvence: Cyklocomputer with Frequency Meter[online]. Brno: Vysoké učení technické, Fakulta elektrotechniky a komunikačních technologií, 2010 [cit. 2014-05-24]. 1 elektronický optický disk [CD-ROM / DVD]. ISBN CYKLOPOČÍTAČ S MĚŘIČEM FREKVENCE. Dostupné z:<https://dspace.vutbr.cz/bitstream/handle/11012/16158/Bakal%C3%A1%C5%99sk%C>



3%A1%20pr%C3%A1ce.pdf?sequence=1. BAKALÁŘSKÁ PRÁCE. Vysoké učení technické v Brně. Vedoucí práce Ing. Jiří Hermány.

- [51] Uctronics.com: GY-30 BH1750FVI intensity Digital Light Sensor Module For Arduino [online]. 2012 [cit. 2014-05-24]. Dostupné z: <http://www.uctronics.com/gy-30-bh1750fvi-intensity-digital-light-sensor-module-for-arduino-p-1494.html>
- [52] Datasheetarchive.com: GP2Y0A21YK [online]. 2005 [cit. 2014-05-24]. Dostupné z: <http://www.datasheetarchive.com/dl/Datasheet-075/DSAE003120.pdf>
- [53] Servodatabase.com: TowerPro MG996R - Standard Servo [online]. 2010 [cit. 2014-05-24]. Dostupné z: <http://www.servodatabase.com/servo/towerpro/mg996r>
- [54] Ebay.com: Mini DC-DC Converter [online]. 2014 [cit. 2014-05-24]. Dostupné z: [http://www.ebay.com/itm/NEW-Mini-DC-DC-Converter-Step-Down-Module-Adjustable-Power-Supply-Output-1-3-17V-/360741066304?pt=LH\\_DefaultDomain\\_0&hash=item53fdd7d640](http://www.ebay.com/itm/NEW-Mini-DC-DC-Converter-Step-Down-Module-Adjustable-Power-Supply-Output-1-3-17V-/360741066304?pt=LH_DefaultDomain_0&hash=item53fdd7d640)
- [55] Camposrc.blogspot.cz [online]. 2011 [cit. 2014-05-24]. Dostupné z: <http://camposrc.blogspot.cz/2012/10/iniciando-no-automodelismo.html>

# Obrázky

1. Obrázok: Teleprezenčný robot v U.S armade [2 ] .....	9
2. Obrázok : Teleprezenčný robot v čínskom armáde [1 ] .....	9
3. Obrázok: Arduino DUE s ARM M3 procesorom [4] .....	10
4. Obrázok : Arduino TRE [4] .....	10
5. Obrázok: Arduino Galileo s Intel procesorom [4] .....	10
6. Obrázok: Arduino ethernet shield [4] .....	11
7. Obrázok: : Bloková schéma jadra ARM Cortex - M3 [6] .....	12
8. Obrázok: Užitočnosti hard a soft realtime systému [8] .....	13
9. Obrázok: Architektura robota [33],[34] .....	14
10. Obrázok: Komunikačný diagram. Zx-znak, Dx-Data .....	16
11. Obrázok: Diagram komunikáciu medzi Simulinkom a Arduinom .....	17
12. Obrázok: Skutočný PWM signál arduina a pomer podľa čísel [12] .....	18
13. Obrázok: Reakcia motora na PWM signál. V hornom grafu sú otáčky motora a dolnom je pomer PWM signálu .....	18
14. Obrázok: Nameraný a teoretický model systému. Modrá odozva je teoretická, čierná odozva je nameraná .....	19
15. Obrázok: Identification toolbox v Matlabe .....	19
16. Obrázok: Nameraný a teoretický model systému. Modrá odozva je teoretická, čierná odozva je nameraná .....	20
17. Obrázok: PI toolbox v Matlabe .....	21
18. Obrázok: Reakcie motora S PI regulátorom .....	21
19. Obrázok: Nelinearita motora .....	22
20. Obrázok: Arduino L298 Shield [14 ] .....	22
21. Obrázok: Diagram funkciu regulátor .....	24
22. Obrázok: Diagram funkciu komunikácia .....	24
23. Obrázok: Diagram funkciu Ultrazvukového senzoru .....	25
24. Obrázok: Diagram funkciu Akcelerometer .....	25
25. Obrázok: Vlákno Regulátora .....	26
26. Obrázok: Vlákno komunikácia .....	26
27. Obrázok: Vlákno infra senzoru .....	26
28. Obrázok: Vlákno svetelného senzora .....	26
30. Obrázok: Vlákno pre poslanie dáta .....	26
29. Obrázok: vlákno akcelerometra .....	26
31. Obrázok: Bloková schéma i2c senzorov .....	27

32. Obrázok: 9DOF senzor s arduinom .....	28
33. Obrázok: BH1750FVI senzor [53].....	29
34. Obrázok: Rýchlosť šírenia zvuku pri zmene teploty[44] .....	30
35. Obrázok: Motor s Hallov senzorom [43].....	30
36. Obrázok: Hallova jev [26] .....	30
37. Obrázok: Princíp proximity s Ir. senzorom [24] .....	31
38. Obrázok: Kamery na robote.....	32
39. Obrázok: Robotické rameno .....	33
40. Obrázok: DC / DC konvertor [54].....	34
41. Obrázok: Li - Po akumulátor [55] .....	34
42. Obrázok: Počítačová aplikácia na riadenie.....	35

## Seznam příloh

Příloha 1. DVD